Best Practices for APIs in REDCap

**Statement from REDCap Developer’s at Vanderbilt:**

*Direct database back-end access in REDCap should not be allowed. Not only does back-end access bypass REDCap’s built-in logging abilities (thus creating compliance issues), but it is likely to cause permanent damage to data. REDCap’s data model is very complex and is always changing, so a simple mistake could cause great damage to valuable data. Additionally, some back-end queries that might be correct now might no longer be correct in a future version of REDCap as the data model changes with every new version. For all these reasons, we (the REDCap developers at Vanderbilt) only recommend using front-end methods in the application (website and API) for extracting data. Any other method is not recommended.*

The only method of interfacing data for UCH REDCap that is supported is the built-in API functionality using the secure API key. This is the only way to comply with the requirements of the Health Insurance Portability and Accountability Act of 1996 (Title 45, Parts 160 and 164, subparts A and E of the Code of Federal Regulation (the “Privacy Rule”). The following is some best practice guidelines for writing APIs for REDCap – and additional best usage practices are at the end of this guide.

1. Use the API Playground if possible to generate your code. Why struggle when a computer can struggle for you?
2. API Export and Import use the same data structure formats. If you have trouble having API Import's data to be formatted properly, do an API Export. Make the data structure viewable in some form via whatever language you want (json\_encode for PHP, JSON.stringify for JavaScript, etc.). Then modify the data values logically to form a new JSON to put into an API Import call.
3. Put the data into a JSON. Do not leave it in its native structure. For some languages, like JavaScript, this is not a big deal. For other languages that do not use JSON-compatible data structures (like PHP, Objective-C, Swift, etc.), putting the native structure into the POST command dooms you to failure. Just throw it into a JSON-encoded structure (which most likely will be a string in standard JSON format), and put it into the data field. I promise, it'll work.
4. API JSON data structures are in arrays ([{ ... }]). If you leave it as a hash ({ ... }), your call will not succeed. Easy to say, easy to forget.
5. API Import calls have a data parameter; API export calls do not. If the content is the same, they are probably the same except for that data parameter. Save yourself a little time.
6. Decode JSONs in PHP using json\_decode(json, true). If you leave off the parameter with the true, PHP will attempt to decode the JSON as a hash; since it is not a hash, it will fail and leave you wondering like you've been stood up on a date. Always, always put the true as the second parameter when dealing with REDCap.
7. Keep in mind memory limitations when the API will be communicating with a phone or tablet. You may need to change the synchronization process to proceed one record at a time. Sometimes you have to break up API Import or Export calls.
8. Any mobile calls are almost certainly cross domain. Turn this on if it is available on your program (as it was available with my on JavaScript's AJAX calls). Read up on JSONP.
9. JSONs easier to handle than XML or CSV. To some extent, this is a personal preference, but JSON parsers seem more friendly than XML or CSV to me for computational processing. Of course, this depends on your application.
10. When using the same basic API code for a new project, duplicate it, then just change the PID and the API Token and you should be good to go.

**REDCap API**

The REDCap API is an interface that allows external applications to connect to REDCap remotely, and is used for programmatically retrieving or modifying data or settings within REDCap. This includes performing automated data imports/exports from a specified REDCap project, importing/exporting a project’s metadata (i.e. data dictionary), events. This is even an API method for creating whole new projects. The API is a built-in feature of REDCap, so no installation is required. The REDCap API implements the use of tokens as a means of authenticating and validating all API requests that are received. Similar to the Data Import Tool in REDCap’s web interface, the API also implements data validation when the API is used for data import purposes in order to ensure that only valid data gets stored. The API provides a very efficient way to move data either to or from another system easily.

REDCap’s API is generally used for three different things:

# PUSHING DATA INTO REDCAP (IMPORTING)

API can be used to populate data into REDCap from other systems, such as electronic medical records, other REDCap projects, web-servers, and statistical programs like [R (https://www.r-project.org/)](https://www.r-project.org/). Complete projects in REDCap can be created and populated with data by using the API. Note: creating projects via the API requires special privileges, talk to your REDCap administrator if you are interested in that type of project.

# PULLING DATA FROM REDCAP (EXPORTING)

Exporting data manually from REDCap is relatively easy. However, you may want to set up an automatic export for data analysis or dashboards. A common use of the API export is for statisticians using R to pull data directly from REDCap into

R. This saves time and enables instantaneous reflection of the most up-to-date data.

# COMMUNICATION BETWEEN A REDCAP SERVER AND THE REDCAP MOBILE APP

Using the API with the REDCap mobile app is the simplest way for users to get acquainted with the API. All you would need is an API token (instructions below). The communication between the server and app is completely automated; all the necessary code is included with the REDCap mobile app.

**GETTING STARTED WITH THE API**

REDCap provides API tokens at the user-project level. This means that if three users on the same project need to use the API, each user will need to individually request an API token. Similarly, if one user wants an API token for three different projects, they will need to request an API token for each project.

1. **USER RIGHTS**

First, you must obtain the appropriate user rights. API user rights are turned off by default for most users. There is a separate user right for each of the main API uses (import, export, and mobile app). Navigate to the user rights menu and check the appropriate boxes for your own user account.

If you do not have access to the user rights menu, ask your PI or designated project manager who does have access to grant you these user rights. The REDCap Admin is not authorized to update individual User Rights.



# GETTING AN API TOKEN

Once you have the correct user rights, you should see the “API” and “API Playground” appear under the applications section. We recommend that you begin with the “API Playground”; thus you can start by clicking on it. REDCap will then prompt you to get an API token. The REDCap Admin will receive your token request and send you the [**API Request Form**.](https://uchredcap.uchc.edu/redcap/surveys/?s=KXWRTPMJK3KTPJKJ)





# LEARN ABOUT THE API “METHODS”

API “Methods” is an IT term for ‘things you can do with the API’. There are currently 20+ different methods. They range

from “importing data” to “exporting project information” and “setting up arms”. It is recommended that users planning to use REDCap’s API have prior coding experience. The REDCap API supports many coding languages, but the two most popular are R and Python. T

There are two ways to learn more about the API methods.

# REDCap API Documentation

API Documentation is similar to a traditional technical instruction manual. If you have programming experience you may be familiar with this style of documentation.

# API Playground

This feature allows you to test the various API methods to see what they do. You can set different parameters for each method and run the resulting code to see the results. In the playground, REDCap generates the appropriate code for a given method in seven different programming languages (PHP, Perl, Python, Ruby, Java, R and cURL) based on your selection. The code can then be copied and pasted into the program of your choice.

The best way to familiarize yourself with the REDCap API is to explore the API Playground.

* 1. Click on the “API Playground” link from the left-hand menu under “Applications.”
	2. Once in the API Playground, there is a blue box with a dropdown menu labeled “API Method.” This dropdown includes all the API actions REDCap can take.
* If a project is in production, the methods listed in this dropdown will be limited so as not to affect real data in the project. This is noted in the green text under the “API Method” dropdown.
	1. Select the method you need from the dropdown menu and complete the additional information. The additional information (e.g., “Format”, “Instrument”, etc.) will vary depending on which API method you choose and the project structure. In the below example, the researcher is asking to export a list of survey participants from the “contact\_info” survey on the “baseline\_survey” event of a longitudinal project as a CSV. a. To see all the API functions REDCap is capable of, and export a .zip file of sample code, click on the “REDCap API documentation” link that is available on both the “API” page and in the “API Playground.”
	2. When you scroll further down the page, there is an open text box with a series of tabs on the top, with each tab corresponding to a coding language. Each tab will provide the API code in the indicated language.



* 1. You can copy and paste this code into the necessary program to run it. Additional data cleaning and manipulation is often necessary. Warning: On the API Playground, there is a button that will let you “Execute Request.” This will perform the API action you are programming and thus affect the data in your project. Use this button with a great amount of caution!



# START USING THE API CODE

After learning about the possibilities of the REDCap API, it’s time to use it! We recommend you start by downloading a free version of R and running an R API method directly from R. Once that is working, you are well on your way to becoming a REDCap API expert. Potential uses of API are limited only by your imagination.

**API Usage Best Practices**

* Never share your API token with anyone. Giving someone else access to your token is a HIPAA violation, and any action taken with your token is logged in REDCap as an action taken by you.
* Do not test API tokens in browsers. Using an API token in plain text within a script is unsecure. An API token should be encrypted within a script, be called via secure environment variables, or otherwise be accessible from the script via other secure mechanisms.
* Before you share code anywhere, remove your API token. This includes sharing code via email, GitHub, etc.
* Regenerate your API token every 90 days, or at any point that you think your token has been compromised. To regenerate your token, go to the API page and select “Regenerate token.” If you are no longer using the API functionality on your project, delete your token.



* Report any security incidents involving your HIPAA token to the REDCap administrators (redcap@uchc.edu) and the IRB and/or your HIPAA liaison.