REDCap API Policy Town Hall

Savas Sevil, REDCap Product Owner
Cate Bauer-Martinez, Senior REDCap Administrator
Priscilla Acquaye, REDCap Administrator
Jing Yang, Applications Support Analyst
Sheersha Kandwal, REDCap Technical Support Analyst

James Chip Masters, Director
Ranjini Kottaiyan, Senior Director
Mark Green, Executive Director
Patricia Kovatch, Professor and Dean

Scientific Computing and Data

Feb 28, 2025

Icahn
School of
Medicine at
Mount
Sinai

Introduction – Team Members

Patricia Kovatch
Professor and Dean for Scientific Computing and Data Div.

Ranjini Kottaiyan
Senior Director Finance & Administration

Farhan Mahmood
Director, Scientific Computing

Eric Rosenberg
System Administrator

Rupan Hossain
Database Administrator

Sheersha Kandwal
REDCap Technical Support Analyst

Mark Green
Executive Director Scientific Computing and Data Div

James Chip Masters
Director of Research Data Services Scientific Computing and Data Div

Savas Sevil
REDCap Product Owner

Cate Bauer-Martinez
Senior REDCap Analyst

Priscilla Acquaye
REDCap Analyst

Jing Yang, PhD
Applications Support Analyst

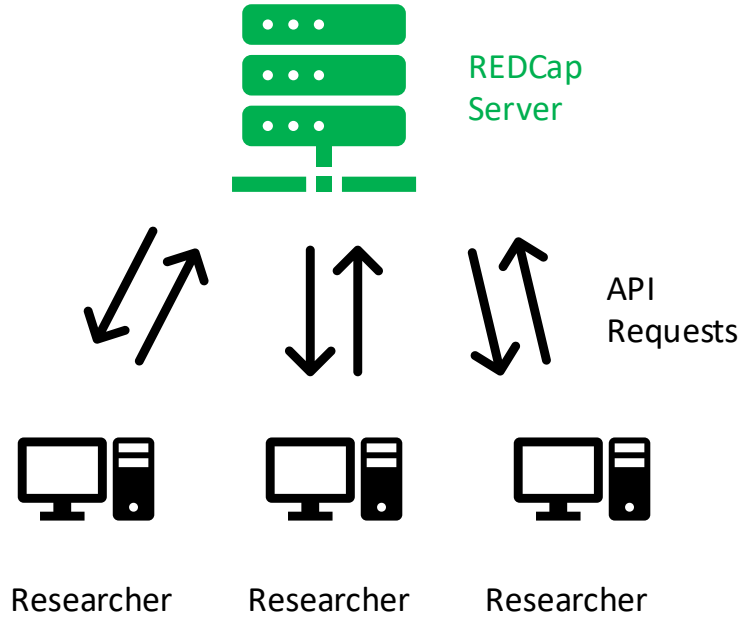
REDCap API Usage Policy

Overview

- About the API
- Current User Statistics
- Existing Policies and Standards
- API Usage Policy
- Future Steps

About the API

- REDCap offers a set of API endpoints for retrieving data from their database
- Some of the endpoints trigger simple queries which are very easy to handle
- Others are more complex and much slower, which can block the ability of others to access services
- We have issued nearly 1000 tokens to enable user code to connect to the API the past few years
- **In the past 12 months: we've had recurring outages due to user code**



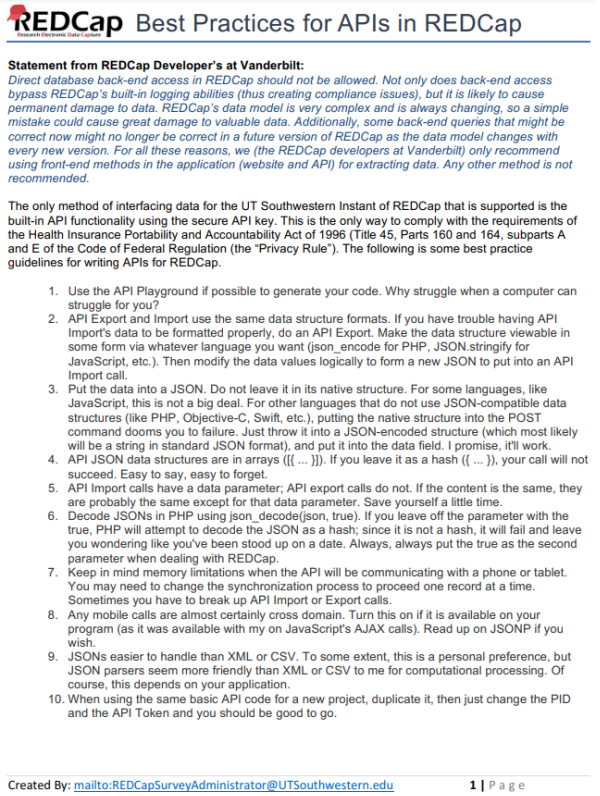
Current State: API User Statistics

- In the past 6 months (since 8/7/25):
  - 236 separate projects
  - 120 unique users
  - Top 10 users responsible for 92% of API calls
  - Top 20 users responsible for 97% of API calls

- We are implementing a new policy to increase REDCap reliability

Best Practices: Existing Policies and Standards

- Several existing standards exist (ex. UT Southwestern)
- There are also several software libraries already written which can help with writing standardized, safe code (ex. PyCap)



REDCap 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 the UT Southwestern Instant of 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.

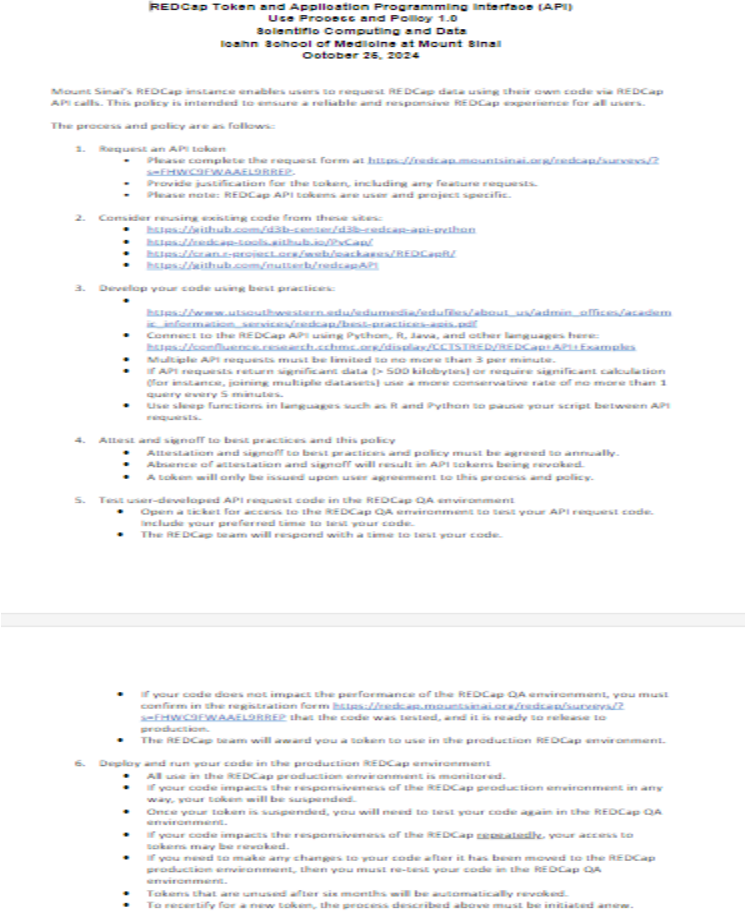
1. Use the API Playground if possible to generate your code. Why struggle when a computer can struggle for you?
2. API Export and Import use the same data structure formats. If you have trouble having API Import's data to be formatted properly, do an API Export. Make the data structure viewable in some 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 if you wish.
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.

Created By: mailto:REDCapSurveyAdministrator@UTSouthwestern.edu 1 | Page

API Usage Policy

- Using existing standards, we drafted a brief policy for API usage based on existing policies in other organizations

- Addressed using best practices suggested by Vanderbilt, using existing coding libraries, testing, throttling requests, monitoring, etc.



REDCap Token and Application Programming Interface (API)
Use Process and Policy 1.0
Scientific Computing and Data
Icahn School of Medicine at Mount Sinai
October 26, 2024

Mount Sinai's REDCap instance enables users to request REDCap data using their own code via REDCap API calls. This policy is intended to ensure a reliable and responsive REDCap experience for all users.

The process and policy are as follows:

1. Request an API token
   - Please complete the request form at https://redcap.mountsinai.org/redcap/surveys/?s=FHWC9FWAAEL9RREP.
   - Provide justification for the token, including any feature requests.
   - Please note: REDCap API tokens are user and project specific.
2. Consider reusing existing code from these sites:
   - https://github.com/d3b-center/d3b-redcap-api-python
   - https://redcap-tools.github.io/PyCap/
   - https://cran.r-project.org/web/packages/REDCapR/
   - https://github.com/nutterb/redcapAPI
3. Develop your code using best practices:
   - https://www.utsouthwestern.edu/edumedia/edufiles/about_us/admin_offices/academic_information_services/redcap/best-practices-apis.pdf
   - Connect to the REDCap API using Python, R, Java, and other languages here: https://confluence.research.cchmc.org/display/KCSTRED/REDCap+API+Examples
   - Multiple API requests must be limited to no more than 3 per minute.
   - If API requests return significant data (> 500 kilobytes) or require significant calculation (for instance, joining multiple datasets) use a more conservative rate of no more than 1 query every 5 minutes.
   - Use sleep functions in languages such as R and Python to pause your script between API requests.
4. Attest and signoff to best practices and this policy
   - Attestation and signoff to best practices and policy must be agreed to annually.
   - Absence of attestation and signoff will result in API tokens being revoked.
   - A token will only be issued upon user agreement to this process and policy.
5. Test user-developed API request code in the REDCap QA environment
   - Open a ticket for access to the REDCap QA environment to test your API request code. Include your preferred time to test your code.
   - The REDCap team will respond with a time to test your code.
   - If your code does not impact the performance of the REDCap QA environment, you must confirm in the registration form https://redcap.mountsinai.org/redcap/surveys/?s=FHWC9FWAAEL9RREP that the code was tested, and it is ready to release to production.
   - The REDCap team will award you a token to use in the production REDCap environment.
6. Deploy and run your code in the production REDCap environment
   - All use in the REDCap production environment is monitored.
   - If your code impacts the responsiveness of the REDCap production environment in any way, your token will be suspended.
   - Once your token is suspended, you will need to test your code again in the REDCap QA environment.
   - If your code impacts the responsiveness of the REDCap repeatedly, your access to tokens may be revoked.
   - If you need to make any changes to your code after it has been moved to the REDCap production environment, then you must re-test your code in the REDCap QA environment.
   - Tokens that are unused after six months will be automatically revoked.
   - To recertify for a new token, the process described above must be initiated anew.

Implementation

- The policy includes user training, agreement to abide by best practices, monitoring, revalidation, revocation policies
- All existing user code needs to be validated in our user environment before usage
- Will phase in over Q1 2025 to groups of users in batches, addressing high API usage researchers first in a targeted fashion
- Will iterate and update the policy as needed

REDCap Token and Application Programming Interface (API) - Use Process and Policy 1.1
Scientific Computing and Data, Icahn School of Medicine at Mount Sinai, February 07, 2025

Mount Sinai's REDCap instance enables users to request REDCap data using their own code via REDCap API calls. This policy is intended to ensure a reliable and responsive REDCap experience for all users.

The process and policy are as follows:

1. Raise a JIRA Ticket at https://scicomp.mssm.edu/jira/servicedesk/customer/portal/9 to request an API Token

- REDCap Admin will provide a link to fill out the Registration Form. Please complete the request form at https://redcap.mountsinai.org/redcap/surveys/?s=FHWC9FWAAEL9RREP.
- Provide justification for the token, including any feature requests.
- Please note: REDCap API tokens are user and project specific.
- Attest and signoff to best practices and this policy
- Attestation and signoff to best practices and policy must be agreed to annually.
- Absence of attestation and signoff will result in API tokens being revoked.
- A token will only be issued upon user agreement to this process and policy.

2. Consider reusing existing code from these sites:
   https://github.com/d3b-center/d3b-redcap-api-python
   https://redcap-tools.github.io/PyCap/
   https://cran.r-project.org/web/packages/REDCapR/
   https://github.com/nutterb/redcapAPI

3. Develop your code using best practices:

   https://www.utsouthwestern.edu/edumedia/edufiles/about_us/admin_offices/academic_information_services/redcap/best-practices-apis.pdf

   Connect to the REDCap API using Python, R, Java, and other languages here: https://confluence.research.cchmc.org/display/CCTSTRED/REDCap+API+Examples

   Multiple API requests must be limited to no more than 3 per minute.

   If API requests return significant data (> 500 kilobytes) or require significant calculation (for instance, joining multiple datasets) use a more conservative rate of no more than 1 query every 5 minutes.

   Use sleep functions in languages such as R and Python to pause your script between API requests.

4. Test user-developed API request code in the REDCap QA environment
   - After completing the Registration Form, please sign up in REDCap QA with your network id and password.
   - REDCap admin will generate an API Token in QA for the username in QA.
   - User is responsible for setting up a test project and testing the code in QA.
   - If your code does not impact the performance of the REDCap QA environment, you must confirm in the registration form https://redcap.mountsinai.org/redcap/surveys/?s=FHWC9FWAAEL9RREP that the code was tested, and it is ready to release to production.
   - The REDCap team will award you a token to use in the production REDCap environment.

5. Deploy and run your code in the production REDCap environment

   All use in the REDCap production environment is monitored.

   If your code impacts the responsiveness of the REDCap production environment in any way, your token will be suspended.

   Once your token is suspended, you will need to test your code again in the REDCap QA environment.

   If your code impacts the responsiveness of the REDCap repeatedly, your access to tokens may be revoked.

   If you need to make any changes to your code after it has been moved to the REDCap production environment, then you must re-test your code in the REDCap QA environment.

   Tokens that are unused after six months will be automatically revoked.

   To recertify for a new token, the process described above must be initiated anew

Appendix - REDCap API Token Registration Form

-Name: _____

- Email: _____

- PI name: _____

- Department: _____

- Project name: _____

- Project id: _____

- Why do you need the token? _____

- Any special features you need _____

- Do any of the following API code repositories meet your needs (please check):
  - [] PyCap (https://redcap-tools.github.io/PyCap/)
  - [] REDCapR (https://cran.r-project.org/web/packages/REDCapR/)
  - [] redcapAPI (https://github.com/nutterb/redcapAPI)
  - [] d3b-redcap (https://github.com/d3b-center/d3b-redcap-api-python)
  - [] I cannot use any of the above

- {Link to training, link to policy}

[] I attest to reading the REDCap API policy, and completed training

Plan for REDCap API rollout:	Target Completion	Status
Q4 2024		
Finalize the API token policy, with active hyperlink endpoints and implement.	11/26/24	Completed
Finalize the REDCap form for “REDCAP API Token Registration” and implement.		Completed
UAT the API request forms, user email, and validate revocation process and procedure.		Completed
Generate and confirm the current API user list with PI and projects.		Completed
Contact API users from last six months to inform them of the API policy and town halls	12/9/24	Completed
Prepare materials for town halls	12/11/24	Completed
Hold town hall #1	12/12/24	Completed
Hold town hall #2	12/17/24	Completed
Prepare plan for montoring and assessing compliance for testing in QA	12/31/24	Completed
Q1 2025		
Update REDCap QA environment to 15.0.11	2/14/2025	Completed
Move QA to Enterprise cluster	3/7/2025	In progress
Divide recent API users into logical groups and schedule them to test in QA	4/2/25 (Tentative)	In progress
Determine how to monitor API and database logs during testing windows and collect telemetry	4/2/25 (Tentative)	In progress
Identify any new API requests and ensure they have agreed to the policy and schedule them to test in QA	4/2/25 (Tentative)	In progress
Conduct testing and monitor	4/2/25 (Tentative)	In progress

Thank you

Scientific Computing and Data

February 28, 2025

Icahn
School of
Medicine at
Mount
Sinai

Appendix