



REDCap API

WHAT IS AN API?

The acronym "API" stands for "Application Programming Interface". An API is just a defined way for external applications to connect to REDCap remotely and is used for programmatically retrieving or modifying data or settings within REDCap, such as performing automated data imports/exports to/from a specified REDCap project. REDCap API is used to make applications, websites, widgets, and other projects that interact with REDCap.

Programs talk to the REDCap API over HTTP, the same protocol that your browser uses to visit and interact with web pages.

API SECURITY: BEST PRACTICES

Although API requests to REDCap are done using SSL (HTTPS), which means that the traffic to and from the REDCap server is encrypted, there is still more that can be done to ensure the highest level of security when using the API. This is especially important if you are moving sensitive data into or out of REDCap.

One thing that is *highly* recommended is for your API script/program (i.e. the thing making the request to the REDCap API) to validate the SSL certificate of the REDCap web server when it makes the API request.

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

HOW TO PREVENT MAN IN THE MIDDLE ATTACKS

Preventing MiM attacks is pretty simple. Essentially all you need to do is to force your API script to validate the SSL certificate of the REDCap server. REDCap's SSL certificate will always be valid, but the hacker's fake certificate can never be determined to be valid if you attempt to validate it. In many programs or programming languages that can make API requests, validating an SSL certificate is often as easy as setting a flag. For example, [cURL](#) is popularly used by many API scripts in programming languages such as PHP, R, SAS, and many more in order to make the web request to REDCap. **So, if your API script is utilizing cURL, all you need to do is modify your script so that it sets the cURL option named CURLOPT_SSL_VERIFYPEER to have a value of TRUE.** Once done, your API script will attempt to make the API request to REDCap *only* if it can validate REDCap's SSL certificate. **Thus, by adding the SSL certificate check, you have completely prevented the**



possibility of MiM attacks and are using the most secure form of communication with the REDCap API. If you are not using cURL, there are plenty of other examples on the web for how to validate an SSL certificate in different programming languages. Such examples can be found simply by Googling the name of your programming language + "verify ssl certificate" (e.g., "[Java verify ssl certificate](#)"), which should provide you with many helpful results.

REMINDER: Please remember that while REDCap itself has many security layers to help protect you and to ensure the highest level of security and data integrity, **it is *your* responsibility to ensure that you are using the most secure methods and best practices when using the REDCap API.**

REHEARSING YOUR API CALLS IN THE API PLAY GROUD Feature in REDCapThe API Playground is an interface that allows experimentation with the REDCap API without actually writing any code. **You can explore all the different API methods and their various options to customize a given API request.** You may even execute a real API request and see the exact response that REDCap returns from the request. If you are interested in creating an API script, the section at the bottom will provide code samples for various programming languages to give you a head start.

Select an API method from the drop-down list below, after which it will load any other options that are specific to that method.

API Method: Export Records

Format: JSON

Type: flat

Records: 1

Fields: record_id, first_name, last_name, address, telephone

Forms: demographics "Basic Demography Form"

Events: event_1_arm_1

Raw Labels: raw

Raw Headers: raw

Checkbox Labels: false

Survey Fields: false

Data Access Groups: false

Filter Logic:

Records Created or Modified in Timespan (begin) (YYYY-MM-DD HH:MM:SS format):

Records Created or Modified in Timespan (end) (YYYY-MM-DD HH:MM:SS format):

Set CSV delimiter character: , (comma) - default

Errors: JSON



Raw Request Parameters

Displayed in the box below are all the POST parameters that would be sent in the API request based on the selections above.

```
content: record
action: export
format: json
type: flat
csvDelimiter:
rawOrLabel: raw
rawOrLabelHeaders: raw
```

Response

Click the Execute Request button to execute a real API request, and it will display the API response in a text box below.

Execute Request

```
[{"record_id": "1", "first_name": "Rock", "last_name": "Scissors", "address": "Rock", "telephone": "(555) 867-5309", "test": "test", "email": "test@noreply.com", "dob": "2024-05-23", "age": "0", "ethnicity": "1", "race": "3", "gender": "0", "height": "130", "weight": "120", "bmi": "71", "comments": "Paper", "demographics_complete": "2"}]
```

HTTP Status: 200

PHP Perl Python Ruby Java R cURL

Displayed in the box below is the code you would use to execute this API request in the selected programming language.

```
<?php
$data = array(
    'token' => '0066328B04C9B9B72D7E1127384CE1BC',
    'content' => 'record',
    'action' => 'export',
    'format' => 'json'
```

ACTIONS SUPPORTED IN FoM REDCAP API (With Current LTS 14.0. *)



Supported Methods

Arms

Export Arms

Import Arms

Delete Arms

Data Access Groups

Export DAGs

Import DAGs

Delete DAGs

Switch DAG

Export User-DAG Assignment

Import User-DAG Assignment

Events

Export Events

Import Events

Delete Events

Field Names

Export List of Export Field Names

Files

Export a File

Import a File

Delete a File

File Repository

Create a New Folder in the File Repository

Export a List of Files/Folders from the File Repository

Export a File from the File Repository

Import a File into the File Repository

Delete a File from the File Repository

Instruments

Export Instruments (Data Entry Forms)

Export PDF file of Instruments

Export Instrument-Event Mappings

Import Instrument-Event Mappings

Logging

Export Logging



Metadata

[Export Metadata \(Data Dictionary\)](#)

[Import Metadata \(Data Dictionary\)](#)

Projects

[Create Project](#)

[Import Project Info](#)

[Export Project Info](#)

[Export Project XML](#)

Records

[Export Records](#)

[Import Records](#)

[Delete Records](#)

[Rename Record](#)

[Generate Next Record Name](#)

Repeating Instruments and Events

[Export Repeating Instruments and Events](#)

[Import Repeating Instruments and Events](#)

Reports

[Export Reports](#)

REDCap

[Export REDCap Version](#)

Surveys

[Export a Survey Link](#)

[Export Survey Participants](#)

[Export a Survey Queue Link](#)

[Export a Survey Return Code](#)

Users & User Privileges

[Export Users](#)

[Import Users](#)

[Delete Users](#)

User Roles

[Export User Roles](#)

[Import User Roles](#)

[Delete User Roles](#)

[Export User-Role Assignment](#)

[Import User-Role Assignment](#)



HTTP STATUS CODES

The REDCap API attempts to return appropriate HTTP status codes for every request. Below is a comprehensive list of all possible HTTP codes that could be returned from the API and what each code signifies.

- **200 OK:** Success!
- **400 Bad Request:** The request was invalid. An accompanying message will explain why.
- **401 Unauthorized:** API token was missing or incorrect.
- **403 Forbidden:** You do not have permissions to use the API.
- **404 Not Found:** The URI you requested is invalid or the resource does not exist.
- **406 Not Acceptable:** The data being imported was formatted incorrectly.
- **500 Internal Server Error:** The server encountered an error processing your request.
- **501 Not Implemented:** The requested method is not implemented.

Error Messages:

When the API returns error messages, it does so in the return format that you specified in your API request (and if not specified, it returns in default format). You can specify the format you want using the **returnFormat** parameter. For example, an error from an XML method might look like this:

```
<?xml version="1.0" encoding="UTF-8" ?>  
<hash>  
  <error>detailed error message</error>  
</hash>
```

If you need any assistance on how to use REDCap API to connect external application, please contact us at **fom.redcap@ubc.ca**.