

# Using Offline Connections

2023-10-26

## Introduction

The `redcapAPI` package works best when the user has permission to export and import data via the REDCap API. This requires that the user to be added to a project, granted API permissions by the project owner, an API token requested, and that the request be approved by the REDCap Administrator. There are many reasons why any one of these steps may not be completed which may leave the user without access to the API.

“Offline connections” are a tool designed to provide the user with at least a subset of the functionality available to API users. Most prominently, `exportRecordsTyped` has a method for offline connections that allows offline users to prepare data for analysis just as they would if an API connection were available. Instead of exporting the various data and meta data elements via the API, the offline connection is constructed using the comma separate value (CSV) files that are downloaded from the REDCap user interface.

## Basic Offline Connections

Offline connections are most commonly made using the CSV downloads from the REDCap project. In order to utilize `exportRecordsTyped`, the two essential downloads are the raw data and the data dictionary. For the best results, the data should be the raw, unlabeled data. Once these two files are downloaded, the offline connection is made by passing the file paths to the `offlineConnection` function. The `redcapAPI` package includes a set of files that may be used for demonstration.

```
suppressPackageStartupMessages({  
  library(redcapAPI)  
})  
  
offline_file_dir <- system.file("extdata/offlineConnectionFiles",  
                                package = "redcapAPI")  
  
records_file <- file.path(offline_file_dir,  
                           "TestRedcapApi_Records.csv")  
metadata_file <- file.path(offline_file_dir,  
                           "TestRedcapAPI_DataDictionary.csv")  
  
off_con <- offlineConnection(meta_data = metadata_file,  
                               records = records_file)  
  
off_con  
  
## REDCap Offline Connection Object  
## Records           : Cached  
## Meta Data         : Cached
```

```

## Arms : Not Cached
## Events : Not Cached
## Instruments : Cached
## Field Names : Cached
## Mapping : Not Cached
## Repeat Inst. : Not Cached
## Users : Not Cached
## User Roles : Not Cached
## Users Role Assignment : Not Cached
## DAGs : Not Cached
## DAG Assigment : Not Cached
## Project Info : Not Cached
## Version : Not Cached
## File Repo : Not Cached

```

With the offline connection object established, the user is now able to prepare data for analysis as if using an API connection.

```
Records <- exportRecordsTyped(off_con)
```

```

## Warning in constructLinkToRedcapForm.redcapOfflineConnection(rcon, form_name =
## Invalid$form_name, : REDCap URL is not stored on the offlineConnection object.
## Links to REDCap forms will not work as desired.

## Warning in .castRecords_attachInvalid(rcon = rcon, Records = Records, Raw =
## Raw, : Some records failed validation. Use 'reviewInvalidRecords' to review the
## failures.

```

```
# Show a subset of the formatted data
head(Records) [1:10]
```

```

##   record_id redcap_event_name redcap_data_access_group redcap_repeat_instrument
## 1          1    event_1_arm_1                      <NA>                  <NA>
## 2          1    event_1_arm_1                      <NA> files_notes_descriptions
## 3          2    event_1_arm_1                      <NA>                  <NA>
## 4          2    event_1_arm_1                      <NA> files_notes_descriptions
## 5          3    event_1_arm_1                      <NA>                  <NA>
## 6          3    event_1_arm_1                      <NA> files_notes_descriptions
##   redcap_repeat_instance treatment email_test letters_only_test
## 1                      <NA>      <NA>          <NA>                  <NA>
## 2                      1      <NA>          <NA>                  <NA>
## 3                      <NA> Treatment dolor@Duis.net          Duis
## 4                      1      <NA>          <NA>                  <NA>
## 5                      <NA> Control  enim@sunt.org          qui
## 6                      1      <NA>          <NA>                  <NA>
##   phone_test           text_test
## 1      <NA>                  <NA>
## 2      <NA>                  <NA>
## 3 (555) 555-2658  voluptate et dolore anim
## 4      <NA>                  <NA>
## 5 (555) 555-2069 culpa in Duis consectetur
## 6      <NA>                  <NA>

```

# Getting Full Functionality from Offline Connections

With just the records and the data dictionary, the offline connection is able to provide the basic functionality of casting records. The casting performed so far, though, yields a warning that the REDCap URL was not provided to the connection object.

As a consequence, links to the REDCap forms for data that failed validation cannot be constructed. In order to get more complete functionality from `exportRecordsTyped`, the user may also provide:

- The REDCap instance URL.
- The REDCap instance version number.
- Events data.
- Project ID number.
- Whether the project is a classical or longitudinal project.

## URL and Version Number

The URL passed to `offlineConnection` may be the same used by the API connection, even if the user does not have permissions to use the API.

The REDCap version number can be found at the bottom of the list of projects on the “My Projects” page.

## Events Data

Events data can be downloaded from the REDCap user interface after opening the “My Projects” page, clicking on “Define My Events”, and then selecting the “Download events (CSV)” option from the “Upload or Download” drop down.

Unfortunately, the events data downloaded from the user interface does not contain the event IDs, which are necessary if links to invalid data forms are desired. If the user wishes to have functional links, they will need to manually provide the event IDs (these can be looked up from the “Define My Events” page). A data frame constructed in the following manner would be suitable to pass to `offlineConnection`.

```
event_data <-  
  read.csv(file.path(offline_file_dir,  
                    "TestRedcapAPI_Events.csv"),  
          stringsAsFactors = FALSE)  
event_data$event_id <- c(427837, 427838)
```

## Project ID and Longitudinal Status

The project ID number and longitudinal status are both values typically found in the project information exported via `exportProjectInformation`. Unfortunately, the user interface does not provide an equivalent download that can be used here. The user will have to provide these values manually to get full functionality from the offline connection.

The project ID number is displayed in the user interface to the right of the project title.

The user can determine if the status is longitudinal by looking at the Project Setup under the “Main Project Settings.” If the “Use longitudinal data collection with defined events?” option is enabled, the project is longitudinal.

The user need not provide the complete project information data frame to get most functionality from the offline connection. the following is enough to gain all functionality provided at the time of this writing.

```
project_info <- data.frame(project_id = 167509,  
                           is_longitudinal = 1)
```

With all of these components in place, the offline connection can now be created with improved functionality. The warnings issued by this call indicate that not all of the fields that *could* be part of the event and project information data have been provided. This is acceptable in this instance. `offlineConnection` is noisy about these types of issues since it cannot make assumptions about the project configuration or how the user intends to utilize the object.

```
off_con <- offlineConnection(meta_data = metadata_file,  
                             records = records_file,  
                             url = "https://redcap.vanderbilt.edu/api/",  
                             version = "13.10.3",  
                             events = event_data,  
                             project_info = project_info)
```

```
## Warning in validateRedcapData(data = .offlineConnection_readFile(events), : The following names in t  
## Downstream functions may not operate as expected. {days_offset, offset_min, offset_max}
```

```
## Warning in validateRedcapData(data = .offlineConnection_readFile(project_info), : The following name  
## Downstream functions may not operate as expected. {project_title, creation_time, production_time, in
```

```
off_con
```

```
## REDCap Offline Connection Object  
## Records : Cached  
## Meta Data : Cached  
## Arms : Not Cached  
## Events : Cached  
## Instruments : Cached  
## Field Names : Cached  
## Mapping : Not Cached  
## Repeat Inst. : Not Cached  
## Users : Not Cached  
## User Roles : Not Cached  
## Users Role Assignment : Not Cached  
## DAGs : Not Cached  
## DAG Assignment : Not Cached  
## Project Info : Cached  
## Version : Cached  
## File Repo : Not Cached
```

This time, when casting records for analysis, the warning about links to invalid data is absent, indicating that `redcapAPI` believes the links are likely to work. These are displayed in the report of invalid records.

```
Records <- exportRecordsTyped(off_con)
```

```
## Warning in .castRecords_attachInvalid(rcon = rcon, Records = Records, Raw =  
## Raw, : Some records failed validation. Use 'reviewInvalidRecords' to review the  
## failures.
```

```
reviewInvalidRecords(Records)
```

## Failed Validations from REDCap project ‘offline’

Thu Oct 26 00:00:00 2023  
Package redcapAPI version 2.8.1  
REDCap version 13.10.3

- Field[number] ‘prereq\_number’ on form ‘branching\_logic’ has 2 failures
  - Row 35, Record Id ‘18’, Value ’ -100’ link
  - Row 37, Record Id ‘19’, Value ’ -100’ link

## Casting Records for Import

Offline connections may also be used to prepare data for import, even if the data will not be imported using the API. For this type of casting, the offline connection only needs the data dictionary. To illustrate, the `Records` object previously prepared may not be prepared for import. After executing `castForImport`, the resulting data frame can be written to a CSV file and uploaded to the project via the REDCap user interface.

```
off_con <- offlineConnection(meta_data = metadata_file)

ForImport <- castForImport(Records,
                           rcon = off_con)
```

## Preparing Data for Offline Users

There arise situations where the data analyst not only doesn’t have access to the API, but may not have the ability to access a project or even REDCap at all. In such instances, the data owner must bear the responsibility to transfer the project data to the analyst in a manner sufficient for the analyst to successfully construct the offline connection object. `redcapAPI` provides tools to assist in this transfer.

The transfer of project data consists of three steps:

1. Export the data and associated meta data from REDCap.
2. Transfer the exported files.
3. Reconstitute the files into an offline connection.

## Exporting Data for Offline Use

Assuming the data owner has full API permissions, the first step may be accomplished using `preserveProject`. This method will export all of the data objects associated with a project (records, data dictionary, events, arms, etc) with options to save the data to either a single `.Rdata` file or a collection of CSV files. (If the data owner does not have API privileges, they will be required to manually download each CSV from the user interface.)

```

save_to_dir = "target_directory"
# will save the object `RedcapList` to a file in the 'save_to_dir' folder
preserveProject(rcon,
                 save_as = "Rdata",
                 dir = save_to_dir)

```

or for a set of CSV files

```

# will save several CSV files to the 'save_to_dir' folder
preserveProject(rcon,
                 save_as = "csv",
                 dir = save_to_dir)

```

## Transferring exported files

redcapAPI does not provide tools to assist in the transfer of files. The data owner and analyst should work together to ensure the files are transferred securely and in accordance with their organizational information security policies. This may involve a shared drive, e-mail, or secure file transfer.

## Reconstitute Data to an Offline Connection

Once the data have been transferred to a location accessible to the data analyst, the offline connection can be made using the `readPreservedProject` function. If the data have been saved to an Rdata file, this is accomplished with

```

path_to_data <- "path on analyst system"

# load the RedcapList object into the environment
load(file.path(path_to_data,
               "[file_name].Rdata"))
off_con <- readPreservedProject(RedcapList)

```

For CSV files,

```

path_to_data <- "path on analyst system"
off_con <- readPreservedProject(path_to_data)

```

## Considerations

redcapAPI uses a rigid naming convention for the Rdata and CSV files generated by `preserveProject`. This rigidity is essential to the ability of `readPreservedProject` to identify saved files with minimal effort from the user. Altering any of the saved file names could prevent the successful reconstruction of the offline connection object.

Furthermore, when working with CSV files, it is assumed that the CSV files in a folder relate to exactly one project. If files for multiple projects are saved in a single folder, `readPreservedProject` will be unable to reconstitute the offline connection object.

## Conclusion

Offline connections are an additional tool for managing REDCap data within R when the user does not have permission to use the API. The most anticipated use of offline connections is to give the user access to `exportRecordsTyped` or `castForImport`. Without permission to use the API, the user should be prepared to manually look up project information in order to use the full power of `redcapAPI`.