Frequently Asked Questions

2023-12-16

Contents

```r
library(redcapAPI)
url <- "https://redcap.vanderbilt.edu/api/" # Our institutions REDCap instance

unlockREDCap(c(rcon = "Sandbox"),
             envir = .GlobalEnv,
             keyring = "API_KEYs",
             url = url)
```

```
## <environment: R_GlobalEnv>
```

Exporting Records

Which is preferred `exportRecordsTyped` or `exportBulkRecords`?

This depends on ones preferred use case. It's important to understand the difference in the two and their relationship.

`exportRecordsTyped` exports a single `data.frame` of all the data requested.

`exportBulkRecords` call `exportRecordsTyped` to create a `data.frame` for each form in the project, or just those requested via the `forms` argument. Additional arguments are all passed to `exportRecordsTyped`. Thus the documentation on validation and casting is the same for both.

If one is starting a new project, we would recommend using `exportBulkRecords` as the subsetting and filtering of empty rows is taken care of, and the user is left with doing the required joins to the data. If one has existing code they are converting that used `exportRecords`, then `exportRecordsTyped` is the recommendation. This is usually followed by code to subset into forms, filtering and then the same joins between these. Thus new projects can save some code by starting with `exportBulkRecords`.

Casting Records

How do I stop casting fields to factors?

I used to be able to set `factors = FALSE` to prevent categorical values from being returned as factors. How do I do that with `exportRecordsTyped`?

Users may substitute an alternate casting list specification within the call to `exportRecordsTyped`. `redcapAPI` provides two lists for this purpose: `default_cast_character` and `default_cast_no_factor`. These two lists are identical and may be used interchangeably.

```
exportRecordsTyped(rcon,
                   cast = default_cast_character)

exportRecordsTyped(rcon,
                   cast = default_cast_no_factor)
```

Aside from not casting factors, all other settings in this list are identical to the default casting.

How do I control the casting of `redcap_event_name`?

In earlier versions of `redcapAPI`, the `redcap_event_name` field commonly returned the values such as `event_1_arm_1`, `event_2_arm_1`, etc. It now returns "fancy" values. How do I get the original behavior?

The `redcap_event_name` field is one of the fields referred to as a "system" field. These fields are not part of the project's data dictionary, and are automatically returned by the API based on the configuration of the project.

By default, `exportRecordsTyped` returns the "labeled" values of the event names.

```
exportRecordsTyped(rcon,
                   fields = "redcap_event_name",
                   records = 1:3)
```

```
##         redcap_event_name
## 1 Event 1 (Arm 1: Arm 1)
## 2 Event 1 (Arm 1: Arm 1)
## 3 Event 1 (Arm 1: Arm 1)
```

This behavior can be changed using the `system` casting override (this will also affect the casting of other system fields).

```
exportRecordsTyped(rcon,
                   fields = "redcap_event_name",
                   records = 1:3,
                   cast = list(system = castRaw))
```

```
##   redcap_event_name
## 1     event_1_arm_1
## 2     event_1_arm_1
## 3     event_1_arm_1
```

Missing Data Detection

Change the Default Missing Data Detection for All Field Types

How do I change the default missing data detection for all field types?

`redcapAPI` has an obscure function that will create a list of overrides for every field type. Use the `na_values` function to create the override list as illustrated below. (Yes, `na_values` takes a function as an argument)

```
customMissingDetection <- function(x, ...){
  is.na(x) | x == "" | x %in% c(-98, -99)
}

Rec <- exportRecordsTyped(rcon,
                          fields = c("days_between",
                                     "days_between_duplicate",
                                     "dropdown_example",
                                     "dropdown_example_duplicate"),
                          na = na_values(customMissingDetection))
Rec
```

```
##   record_id days_between days_between_duplicate dropdown_example
## 1         1           10                     10         One week
## 2         2           22                     22      Three weeks
## 3         3           NA                     NA             <NA>
## 4         4           NA                     NA             <NA>
##   dropdown_example_duplicate
## 1                   One week
## 2                Three weeks
## 3                       <NA>
## 4                       <NA>
```