# Synapse Demo - Table Uploads

Aryton Tediarjo

2021-03-17

This R markdown is used for showcasing basic interaction with Synapse using RStudio. It will cover on how to utilize our Synapse R client to create table entity, which is a SQL query-able tables in Synapse interface.

Full Documentation: https://r-docs.synapse.org/articles/tables.html

## Import Libraries

```
library(data.table)
library(tidyverse)
library(synapser)
library(synapserutils)
synLogin()
```

```
## Welcome, Aryton Tediarjo!
```

```
## NULL
```

## Global Variables

```
SYNAPSE_PROJECT_ID <- "syn20816722" # where you will be saving all your files
FILEPATH_1 <- file.path("data", "sample_2.tsv")
FILEPATH_2 <- file.path("data", "sample_3.tsv")
TABLE_RESULTS <- "Synapse Demo Table"
```

```
data <- fread(FILEPATH_1) %>% tibble::as_tibble()
data
```

```
## # A tibble: 5 x 4
##   user       createdOn        x      y
##   <chr>      <date>       <dbl>  <dbl>
## 1 2ujCqOOTaL 2020-01-26 -0.0925 -2.06
## 2 ZjFIjuDMRu 2020-07-09  0.638  -1.06
## 3 OW71lKO2MZ 2020-10-17  0.202  -1.16
## 4 gyh0Xpo5Ex 2020-11-09 -0.0699  1.65
## 5 crmp5Sbvy2 2020-11-26 -0.713   0.449
```

### QnD storing table

A quick way to store table is to use available wrapper from Synapse R Client. It will infer data type of each column in the schema.

```
#' wrapper create table function
table <- synapser::synBuildTable(
    TABLE_RESULTS, SYNAPSE_PROJECT_ID, data
)

#' store to synapse
table <- synStore(table)
```

```
## Uploading [--------------------]0.00%   0.0bytes/351.0bytes  file2c956e501add     Uploading [#######
```

### Custom Columns Types

A more manual way is to define each of the column schema individually.

```
#' delete table
synapser::synDelete(table$tableId)
```

```
## NULL
```

```
#' set custom columns
cols <- list(
    Column(name = "user", columnType = "STRING", maximumSize = 20),
    Column(name = "createdOn", columnType = "DATE"),
    Column(name = "x", columnType = "DOUBLE"),
    Column(name = "y", columnType = "DOUBLE"))

# set schema entity
schema <- Schema(name = TABLE_RESULTS,
                 columns = cols,
                 parent = SYNAPSE_PROJECT_ID)

#' recreate table entity
table <- Table(schema, data)
table <- synStore(table)
```

```
## Uploading [--------------------]0.00%   0.0bytes/351.0bytes  file2c9552c00021     Uploading [#######
```

### Appending new rows to Synapse Tables

Appending new data can be achieved by storing a new data (given matching columns) to the previous table id.

```
#' what is the table Id?
tableId_in_synapse <- table$tableId
```

```
#' read data for appending
append_data <- fread(FILEPATH_2) %>%
    tibble::as_tibble()

#' append to table
synStore(Table(tableId_in_synapse, append_data))
```

```
## Uploading [--------------------]0.00%   0.0bytes/346.0bytes  file2c955af40650     Uploading [########
```

```
## <synapseclient.table.CsvFileTable object at 0x124314550>
```

**Querying Data Examples**

```
cols <- c("data.json")
tbl_entity <- synTableQuery("SELECT * FROM syn25173598")
tbl_data <- tbl_entity$asDataFrame() %>%
  tidyr::pivot_longer(cols = all_of(cols),
                      names_to = "column",
                      values_to = "fileHandleId")

tbl_files <- synDownloadTableColumns(tbl_entity, columns = cols) %>%
    tibble::enframe(.) %>%
    dplyr::select(fileHandleId = name, filePath = value)
```

```
## Downloading 0 files, 23 cached locally
```

```
result <- tbl_data %>%
  dplyr::inner_join(tbl_files, by = c("fileHandleId"))


result %>%
    dplyr::select("recordId", "healthCode", "createdOn","filePath") %>%
    slice(1:5)
```

```
## # A tibble: 5 x 4
##   recordId      healthCode     createdOn           filePath
##   <chr>         <chr>          <dttm>              <chr>
## 1 f0CE2TtdPsec~ r5JD3c772ffRd~ 2021-03-11 16:09:36 /Users/atediarjo/.synapseCac~
## 2 gxiO-bvLHYku~ r5JD3c772ffRd~ 2021-03-11 16:36:39 /Users/atediarjo/.synapseCac~
## 3 LmiRQNhKF8u2~ sX90SvP79hmnt~ 2021-03-13 01:31:26 /Users/atediarjo/.synapseCac~
## 4 gg7N3WBA26M8~ sX90SvP79hmnt~ 2021-03-13 01:30:06 /Users/atediarjo/.synapseCac~
## 5 z0F3f1mMAcqr~ sX90SvP79hmnt~ 2021-03-13 01:38:49 /Users/atediarjo/.synapseCac~
```

**Extras: Useful code snippet to parse file attachment in Synapse Table**

```
#' Function to parse file attachment
#' @param synId the synapse table Id
#' @param cols the column of file attachment in table
```

```r
parse_file_attachment <- function(synId, cols){
  tbl_entity <- synTableQuery(glue::glue("SELECT * FROM {synId}"))
  tbl_data <- tbl_entity$asDataFrame() %>%
    tidyr::pivot_longer(cols = all_of(cols),
                        names_to = "column",
                        values_to = "fileHandleId")

  tbl_files <- synDownloadTableColumns(tbl_entity, columns = cols) %>%
    tibble::enframe(.) %>%
    dplyr::select(fileHandleId = name, filePath = value)

  #' printing head and filepaths
  tbl_data %>%
    dplyr::inner_join(tbl_files, by = c("fileHandleId"))
}

parse_file_attachment(synId = "syn25173598",
                      cols = c("data.json")) %>%
    dplyr::select("recordId", "healthCode", "createdOn","filePath") %>%
    slice(1:5)
```

```
## Downloading 0 files, 23 cached locally
```

```
## # A tibble: 5 x 4
##   recordId      healthCode     createdOn           filePath
##   <chr>         <chr>          <dttm>              <chr>
## 1 f0CE2TtdPsec~ r5JD3c772ffRd~ 2021-03-11 16:09:36 /Users/atediarjo/.synapseCac~
## 2 gxiO-bvLHYku~ r5JD3c772ffRd~ 2021-03-11 16:36:39 /Users/atediarjo/.synapseCac~
## 3 LmiRQNhKF8u2~ sX90SvP79hmnt~ 2021-03-13 01:31:26 /Users/atediarjo/.synapseCac~
## 4 gg7N3WBA26M8~ sX90SvP79hmnt~ 2021-03-13 01:30:06 /Users/atediarjo/.synapseCac~
## 5 z0F3f1mMAcqr~ sX90SvP79hmnt~ 2021-03-13 01:38:49 /Users/atediarjo/.synapseCac~
```