Hydrocron Example Lake Data Download

Mike Talbot

2025-02-01

This notebook downloads and parses the lake level time series data in the Hydrocron example: https://podaac.github.io/hydrocron/examples.html

It's basically a minimal example of how the API works and how the JSON can be parsed in R.

You can view the lakes data at the link below. The relevant information needed for the API request is simply the Lake ID (replace the `lake_id` variable in the script below). You can also download data directly through the viewer.

https://apps.usgs.gov/wisp?lat=40.579328&lon=-105.095101&zoom=10.89&basemap=%22Light%22&useDevStyles=false&activeLayers=%5B%7B%22id%22%3A%22lakes%22%7D%5D

Download

```r
# Define download directory and file name
out_dir <- "data"

# Define an output file name
out_file <- "example_data.csv"

# Define the ID of the lake you want to query (7420101733 is Horsetooth Reservoir)
lake_id <- "7420101733"

# Define the start and end times of the time series of lake levels
start_time <- "2023-01-31T00:00:00Z"
end_time <- "2025-01-31T00:00:00Z"

# Define the example API URL
url <- str_glue("https://soto.podaac.earthdatacloud.nasa.gov/hydrocron/v1/timeseries?feature=PriorLake&

# Make the GET request to the API
response <- GET(url)

# Check the status of the response
if (status_code(response) == 200) {
  # Parse the JSON response
  json_content <- content(response, as = "text", encoding = "UTF-8")
  parsed_json <- fromJSON(json_content)

  # Extract the CSV data from the JSON
  csv_content <- parsed_json$results$csv

  # Write the extracted data to a CSV file
  write_file(csv_content, file.path(out_dir, out_file))
  cat("Data downloaded successfully.\n")
} else {
  cat("Failed to download data. Status code:", status_code(response), "\n")
}
```

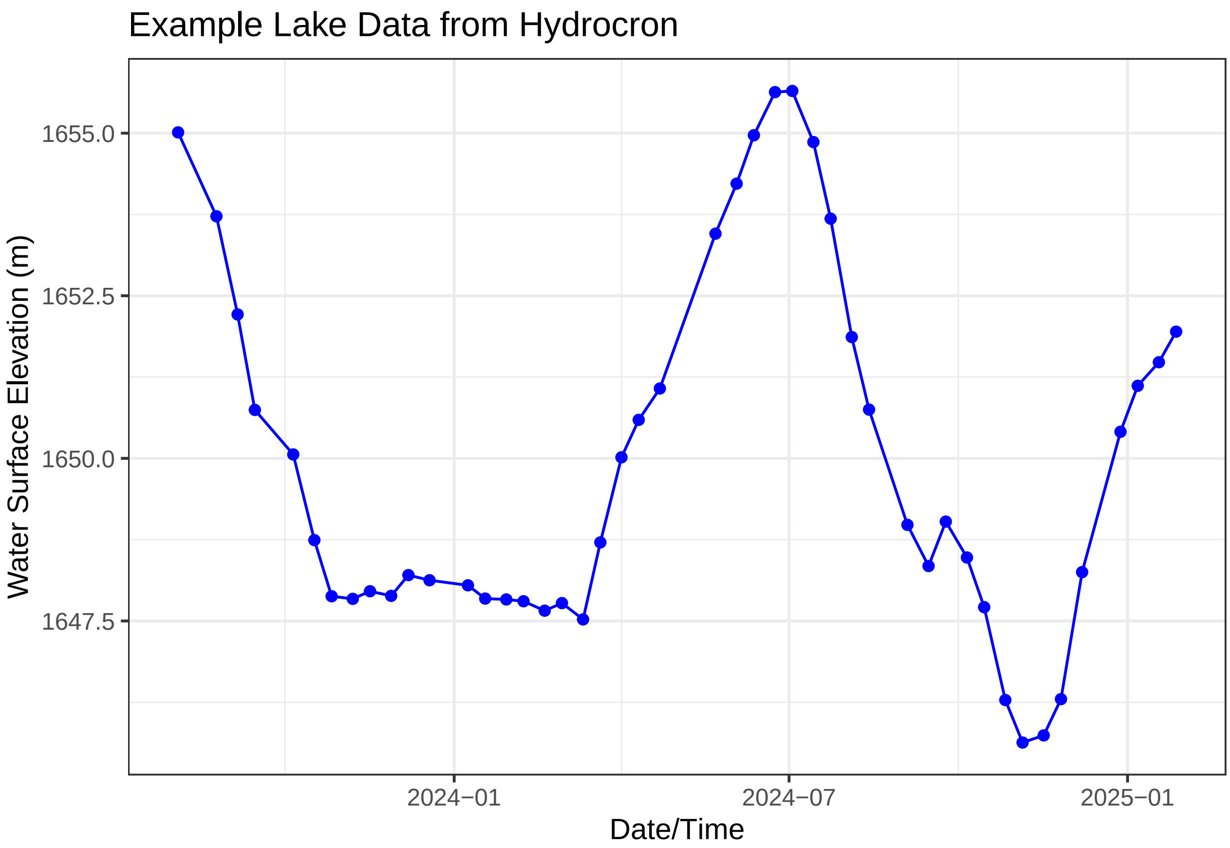
```
## Data downloaded successfully.
```

Plot

```r
# Read data
data <- read_csv(file.path(out_dir, out_file), show_col_types = F)

# View first six rows ("head") of data
print(head(data))
```

```
## # A tibble: 6 x 11
##       lake_id time_str         wse area_total quality_f collection_shortname crid
##         <dbl> <chr>          <dbl>      <dbl>     <dbl> <chr>                <chr>
## 1 7420101733 2023-08-0~  1.66e 3    7.60e 0         0 SWOT_L2_HR_LakeSP_2~ PGC0
## 2 7420101733 no_data    -1.00e12   -1.00e12      -999 SWOT_L2_HR_LakeSP_2~ PGC0
## 3 7420101733 2023-08-2~  1.65e 3    7.46e 0         0 SWOT_L2_HR_LakeSP_2~ PGC0
## 4 7420101733 no_data    -1.00e12   -1.00e12      -999 SWOT_L2_HR_LakeSP_2~ PGC0
## 5 7420101733 no_data    -1.00e12   -1.00e12      -999 SWOT_L2_HR_LakeSP_2~ PGC0
## 6 7420101733 2023-09-0~  1.65e 3    7.15e 0         0 SWOT_L2_HR_LakeSP_2~ PGC0
## # i 4 more variables: PLD_version <dbl>, range_start_time <dttm>,
## #   wse_units <chr>, area_total_units <chr>
```

```r
# Filter out no_data values
data <- data %>%
  filter(time_str != "no_data")

# Convert `time_str` to a datetime object
data <- data %>%
  mutate(time_str = ymd_hms(time_str))

# Plot data
ggplot(data, aes(x = time_str, y = wse)) +
  geom_point(col = "blue") +
  geom_line(col = "blue") +
  xlab("Date/Time") +
  ylab("Water Surface Elevation (m)") +
  ggtitle("Example Lake Data from Hydrocron") +
  theme_bw()
```


Example Lake Data from Hydrocron
Water Surface Elevation (m)
1655.0
1652.5
1650.0
1647.5
2024−01
2024−07
2025−01
Date/Time