

logging.py

using requests to collect api data from
CU's bike logs & webscraping to
collect CU weather data

logging.py

api

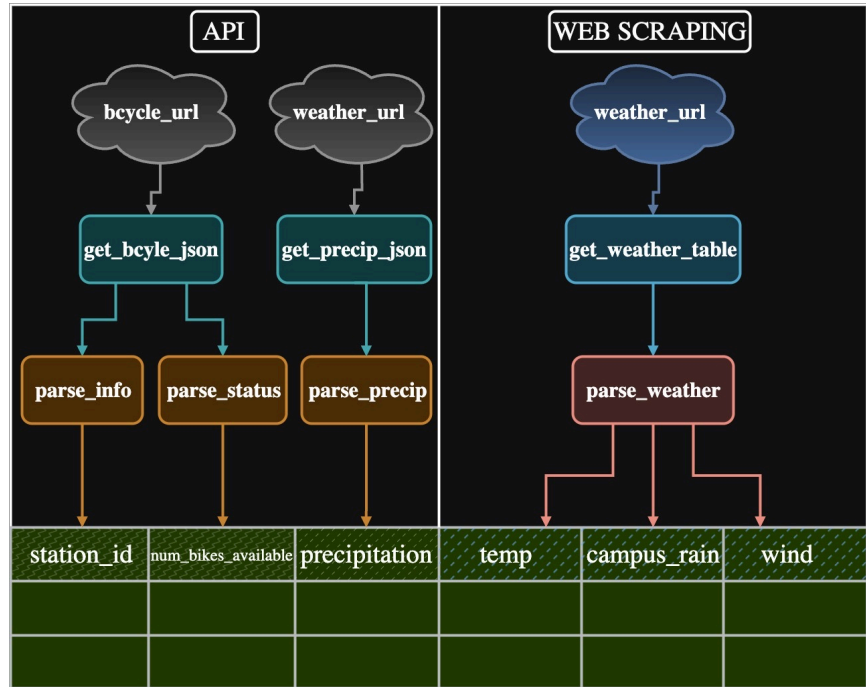
using `get_bcycle_json`, we can make get requests for specific jsons. from that specific json we chose, we will either use `parse_info` or `parse_status` function. those functions allow for us to handpick the values we want.

web scraping

using `get_weather_table`, we can scrape the website for the table with the weather info.

with the given weather table, we can use `parse_weather` function.

this function will allow us to handpick the weather values we want



api logging

Main URL

the `bicycle_url` used is the GBFS (General Bikeshare Feed Specification) for the boulder bicycle stations. these are organized in json format.

Array Organization

all arrays are either 1d or 2d numpy arrays.

`request_names = [str,...]`

this array holds the two get request names for the url.

`station_array = [int,...]`

this array holds the index of the bicycle stations on campus.

`info_columns = [str,...]`

this array holds the values of the specific data wanted from the info json.

`status_columns = [str,...]`

this array holds the values of the data we want to collect from the status json.

get_bicycle_json

this function takes in a string as the url with the `request_name`. this is fetched as a get request, which will return a json of the specific request.

parse_info

this function takes in the `info` json from `get_bicycle_json()`.

using json parsing we categorize our search by only indexing the values in the `station_array`.

then using the same technique, we categorize our search by indexing the values in our `info_columns`.

these values are then converted to dataframe columns that represent the `info_columns`.

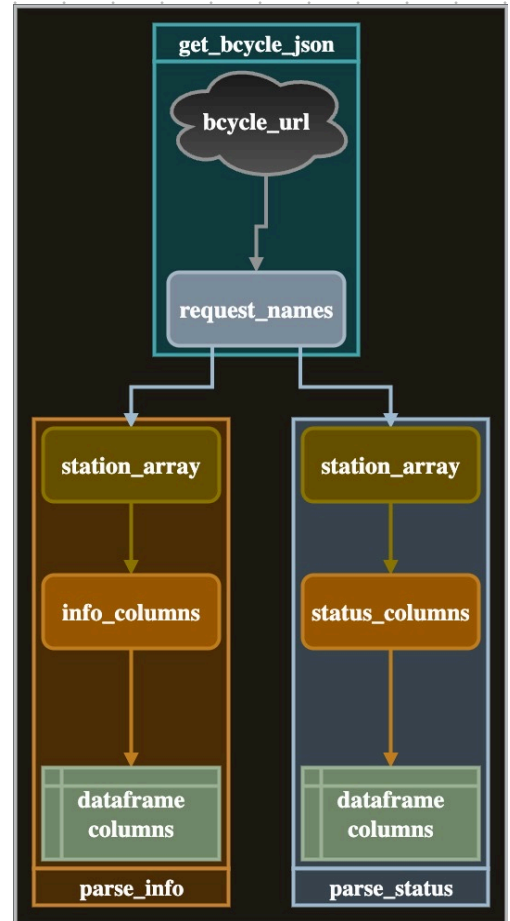
parse_status

this function takes in the `status` json from `get_bicycle_json()`.

using json parsing we categorize our search by only indexing the values in the `station_array`.

then using the same technique, we categorize our search by indexing the values in our `status_columns`.

these values are then converted to dataframe columns that represent the `status_columns`.



dataframe organization

the columns we are collecting for the data frame:

`station_id` `name` `lon` `lat` `num_docks_available` `num_bikes_available`

each value is saved to the df using `df.at`