Notebook_Anqi Li

Anqi Li

14/02/2022

2022-02-14

1. What I did:

- Had a meeting with other group members, decided our target and devision for the work.
- Implemented the query function which can return an object with useful information in a nice format. Tested it with a simple usecase. (According to the guideline here: https://httr.r-lib.org/articles/api-packages.html.)

- First, created the query url with the parameters specified in the API document.

```
base_url <- "http://api.weatherapi.com/v1/history.json?"
  full_url <- POST(base_url, query = list(key=key,q=q,dt=dt),encode = "raw")
  path<-paste0("key=",key,"&q=",q,"&dt=",dt)
  resp <- GET(full_url)
```

- Dealed with the error here: if the return format is not json.

```
if (http_type(resp) != "application/json") {
    stop("API did not return json", call. = FALSE)
  }
```

- Parse the returned response to json.

```
parsed <- jsonlite::fromJSON(content(resp, as = "text", encoding = "UTF-8"), simplifyVector = FALSE)
```

- Handled the response error. Here I designed the output as a well-defined format: HTTP Status Code, Error code, and description.

```
if (http_error(resp)) {
    stop(
      sprintf(
        "GitHub API request failed [%s]\n%s\n<%s>",
        status_code(resp),
        parsed$error$message,
        parsed$error$code
      ),
      call. = FALSE
    )
  }
```

- return a helpful object: designed a list with 4 data attributes(location_data, daily_data, astro_data, hourly_data) and 3 information attributes(path, response, class). The 4 data attributes contain all the information that can get from the "history" part of the weather api. I parsed and converted them into proper format. All of them are lists that contains readable information. The 3 information attributes contains the information about the query: the path used for the query, the response, and the class I

designed.

```
structure(
    list(
      location_data= parsed[[1]],
      daily_data = parsed[[2]]$forecastday[[1]]$day,
      astro_data= parsed[[2]]$forecastday[[1]]$astro,
      hourly_data=parsed[[2]]$forecastday[[1]]$hour,
      path = path,
      response = resp,
      class = "weather_api_history"
    )
  )
```

- I used the query to made a simple useful use case demo without errors.

```
res<-weather_api_history("##samplekeyhere##","London","2022-02-08")
#(1)data
#location data
res$location_data
#daily temp summary
res$daily_data
#astro data
res$astro_data
#hourly data
length(res$hourly_data)#a list
res$hourly_data[[1]]
#(2)information
#path
res$path
#response
res$response
#class
res$class
```

2. Its role in the bigger group picture: The "history" part is one out of the 3 critical parts we are going to query.

3. Link to github commit: https://github.com/xintian927/534-proejct/commit/5f0d6356411705a902f02411c9bbf6824816