臺灣周邊海域

漁場環境監測資料 API 介接

暨

R 語言讀取說明

V1.2

2024/12/05

目錄

前言 3

R API 操作說明 4

一、 讀取 API 資訊 4

二、 將資料轉為 Data Frame 6

API Code 全覽 11

前言

API 是一種軟體工具，主要功能是作為後端資料庫後台與前端使用者之間的橋樑，且前端不限於原始操作介面，這樣的特性使其可以於不同的應用程式間進行資訊的交流，並將複雜的功能系統包裝為簡易使用的方式，供開發人員及應用程式使用。

通過 API，應用程式可以請求後台執行特定的動作，或請求特定的資料，例如擷取數據、發送數據、執行特定功能等。API 也可以支援多種協議和格式，例如 HTTP、REST、JSON 和 XML 等，這使得不同平台和程式語言之間的整合變得更容易。利用 API，只需將相關資訊提供(如經授權的 API key)給使用者，使用者便可以根據需求取得相對應的資料

本次專案針對臺灣周邊海域漁場環境監測資料管理系統建立 API 乙式，並針對 API 金鑰創立金鑰管理系統乙式。此 API 得以使授權使用者利用 R 語言或其他程式語言進行資料篩選，並加以讀取資料，以利於其他國內資料庫與水試所進行資料的介接。第一階段預計開放葉綠素甲、溫度及鹽度資料。

若您已成功向農業部水產試驗所申請 API 金鑰，即可參照下列程式進行介接。

R API 操作說明

一、讀取 API 資訊

1. 先利用 install.packages()安裝 tidyr、httr 與 rvest 的函數集

```
> install.packages("tidyr","httr","rvest")
```

2. 再用 library()載入 tidyr、httr 與 rvest 的函數集。

```
> library(tidyr)
> library(httr)
> library(rvest)
```

3. 在讀取 API 前先指定要取得資料的 API 網址

```
> url = "http://140.110.11.146/oceandb/OceandbAPI.php"
```

4. 利用 GET() 函示讀取資料，

```
> df = GET(url = url,
           add_headers(Authorization = "token"),
           query = list(StartDate = "2023-01-01",
                        EndDate = "2023-12-31",
                        Station = 1,
                        Datatype = "TEMPERATURE"))
```

其中參數包含：

參數名稱	內容
url	API 網址
token	API 金鑰
StartDate	資料開始日期(YYYY/MM/DD)，若無輸入則預設為從最一開始起
EndDate	資料結束日期(YYYY/MM/DD)，若無設定則預設為至最後一筆
Station	測站，若無設定則預設為全部，對應 MySQL 中的 STATION_ID
Datatype	資料種類，目前開放 TEMPERATURE(溫度)、SALINITY(鹽度)及、CHLOROPHYLL(葉綠素甲)開放使用，若無設定則顯示錯誤。

二、將資料轉為 Data Frame

1. 先利用 rawToChar 將 df 中含有資料的部分轉換成文字，

```
> df2 = rawToChar(df$content)
```

轉換完的資料會全部以文字呈現，並用跳脫字元 "\n" 將每一行的資料區分，如下圖所示：

```
> print(df2)
[1]"SERIAL,CRUISE,STATION,DEPTH,BIG,SMALL,TOTAL\n1,202202,1,5,0
.0696421,0.2097,0.2793\n2,202202,1,25,0.0812909,0.1447,0.226\n3
,202202,1,50,0.0387426,0.1024,0.1411\n4,202202,1,75,0.0169504,0
.0248,0.0417\n5,202202,1,100,0.0141759,0.0575,0.0717\n6,202202,
1,150,0.0199549,0.0278,0.0478\n7,202204,1,5,0.275,2.096,2.371\n
8,202204,1,25,0.736,0.645,1.381\n9,202204,1,50,0.666,1.037,1.70
3\n10,202204,1,75,0.495,0.946,1.441\n11,202204,1,100,0.454,2.18
3,2.638\n12,202204,1,150,0,0,0\n13,202208,1,5,0.029,0.035,0.064
\n14,202208,1,25,-0.008,0.105,0.097\n15,202208,1,50,-
0.016,0.028,0.012\n16,202208,1,75,0.023,0.03,0.053\n17,202208,1
,100,0.007,0.009,0.015\n18,202208,1,150,0.004,0.003,0.007\n19,2
02211,1,5,0.005617,0.014741,0.020358\n20,202211,1,25,0.00300265
,0.0351056,0.0381083\n21,202211,1,50,0.00440333,0.0138092,0.018
2125\n\n"
```

2. 因此，要先利用 strsplit() 將每一列的資料分開，並將其轉換為 dataframe 形式，並將欄位名稱定義為 "temp"

```
> lines = as.data.frame(strsplit(df2,"\n")[[1]])
```

分開後的資料會如下圖所示：

```
> print(lines)
temp
1 SERIAL,CRUISE,STATION,DEPTH,BIG,SMALL,TOTAL
2 1,202202,1,5,0.0696421,0.2097,0.2793
3 2,202202,1,25,0.0812909,0.1447,0.226
4 3,202202,1,50,0.0387426,0.1024,0.1411
5 4,202202,1,75,0.0169504,0.0248,0.0417
6 5,202202,1,100,0.0141759,0.0575,0.0717
7 6,202202,1,150,0.0199549,0.0278,0.0478
8 7,202204,1,5,0.275,2.096,2.371
9 8,202204,1,25,0.736,0.645,1.381
10 9,202204,1,50,0.666,1.037,1.703
11 10,202204,1,75,0.495,0.946,1.441
12 11,202204,1,100,0.454,2.183,2.638
13 12,202204,1,150,0,0,0
14 13,202208,1,5,0.029,0.035,0.064
15 14,202208,1,25,-0.008,0.105,0.097
16 15,202208,1,50,-0.016,0.028,0.012
17 16,202208,1,75,0.023,0.03,0.053
18 17,202208,1,100,0.007,0.009,0.015
19 18,202208,1,150,0.004,0.003,0.007
20 19,202211,1,5,0.005617,0.014741,0.020358
21 20,202211,1,25,0.00300265,0.0351056,0.0381083
22 21,202211,1,50,0.00440333,0.0138092,0.0182125
```

3. 此時資料依舊是呈現列表的形式，因此需要利用 separate() 函式將列表資料內的內容以逗號分割成各別的資料內容，但因為在分割時需要指定資料表的欄位數，所以先再次使用 strsplit() 將第一列的欄位名稱分割出來以作為欄位數量的參考，如下圖所示：

```
> name = strsplit(lines[[1]], ",")[[1]]
```

此時就可以將欄位名稱分隔出來。

```
> print(name)
[1] "SERIAL" "CRUISE" "STATION" "DEPTH" "BIG" "SMALL" "TOTAL"
```

4. 完成後我們就可以利用 separate() 將資料分割完成

```
> names(lines)="temp"
> dat = separate(lines, temp, name, sep=",")
> print(dat)
```

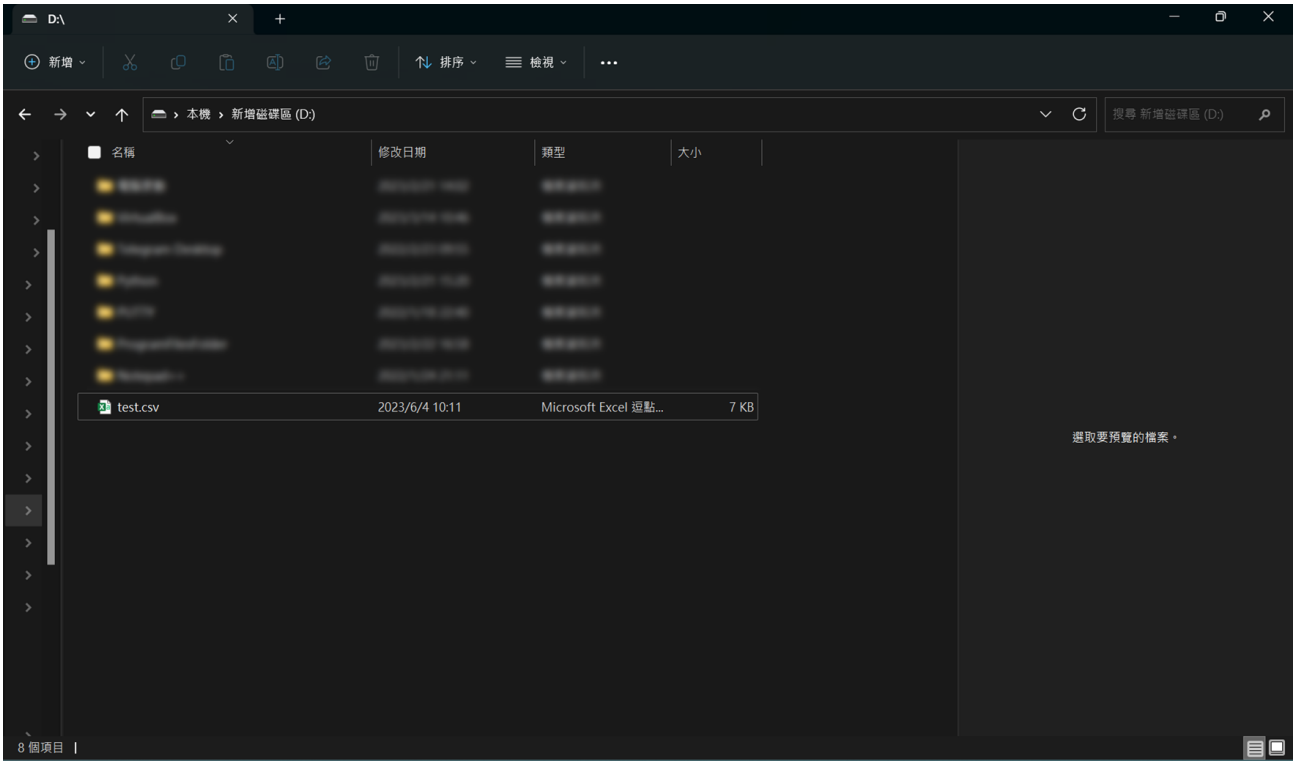
	SERIAL	CRUISE	STATION	DEPTH	BIG	SMALL	TOTAL
1	SERIAL	CRUISE	STATION	DEPTH	BIG	SMALL	TOTAL
2	1	202202	1	5	0.0696421	0.2097	0.2793
3	2	202202	1	25	0.0812909	0.1447	0.226
4	3	202202	1	50	0.0387426	0.1024	0.1411
5	4	202202	1	75	0.0169504	0.0248	0.0417
6	5	202202	1	100	0.0141759	0.0575	0.0717
7	6	202202	1	150	0.0199549	0.0278	0.0478
8	7	202204	1	5	0.275	2.096	2.371
9	8	202204	1	25	0.736	0.645	1.381
10	9	202204	1	50	0.666	1.037	1.703
11	10	202204	1	75	0.495	0.946	1.441
12	11	202204	1	100	0.454	2.183	2.638
13	12	202204	1	150	0	0	0
14	13	202208	1	5	0.029	0.035	0.064
15	14	202208	1	25	-0.008	0.105	0.097
16	15	202208	1	50	0.016	0.028	0.012
17	16	202208	1	75	0.023	0.03	0.053
18	17	202208	1	100	0.007	0.009	0.015
19	18	202208	1	150	0.004	0.003	0.007
20	19	202211	1	5	0.005617	0.014741	0.020358
21	20	202211	1	25	0.00300265	0.0351056	0.038108
22	21	202211	1	50	0.00440333	0.0138092	0.018213
23		<NA>	<NA>	<NA>	<NA>	<NA>	<NA>

5. 利用 write.csv() 可以將資料輸出成 .csv 格式

```
> path = "D:/test.csv"
> write.csv(dat, path)
```

dat 是想要輸出的資料，path 是要輸出的路徑及資料要存檔的名稱

6. 輸出成功即可檢視資料



API Code 全覽

```
> install.packages("tidyr","httr","rvest")

> library(tidyr)
> library(httr)
> library(rvest)

> url = "http://140.110.11.146/oceandb/OceandbAPI.php"

> df = GET(url = url,
           add_headers(Authorization = "XXXX-1234-ABCD-5678"),
           query = list(StartDate = "2023-01-01",
                        EndDate = "2023-12-31",
                        Station = 1,
                        Datatype = "TEMPERATURE"))

> df2 = rawToChar(df$content)
> lines = as.data.frame(strsplit(df2,"\n")[[1]])
> names(lines)="temp"
> name = strsplit(lines[[1]], ",")[[1]]
> dat = separate(lines, temp, name, sep=",")
> path = "D:/test.csv"
> write.csv(dat, path)
```