Getting Data from the Web

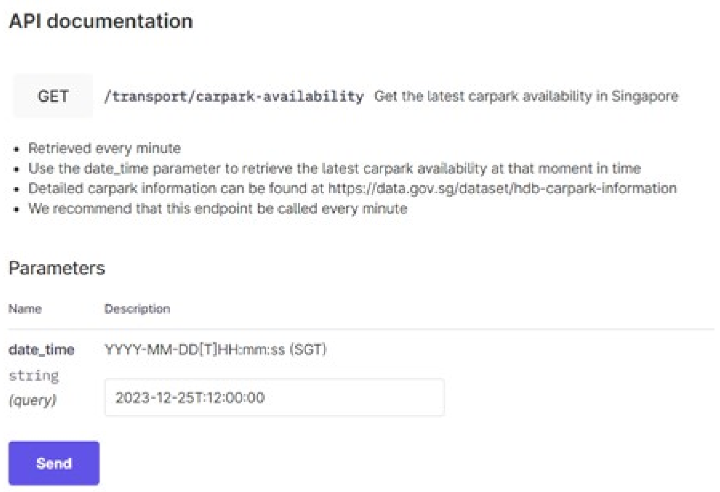
Pre-requisite: JSON file format, HTTP GET requests

API Handling

There are many ways in which API data can be retrieved and handled. The example below illustrates how data can be retrieved using API on data.gov.sg. Take note that the code to retrieve from other API websites might differ accordingly.

We would like to get the carpark availability in Singapore on Christmas day noon in 2023. The data API documentation can be found in https://beta.data.gov.sg/collections/85/view.

1. Input "2023-12-25T:12:00:00" into the date_time field click Send.



API documentation

GET `/transport/carpark-availability` Get the latest carpark availability in Singapore

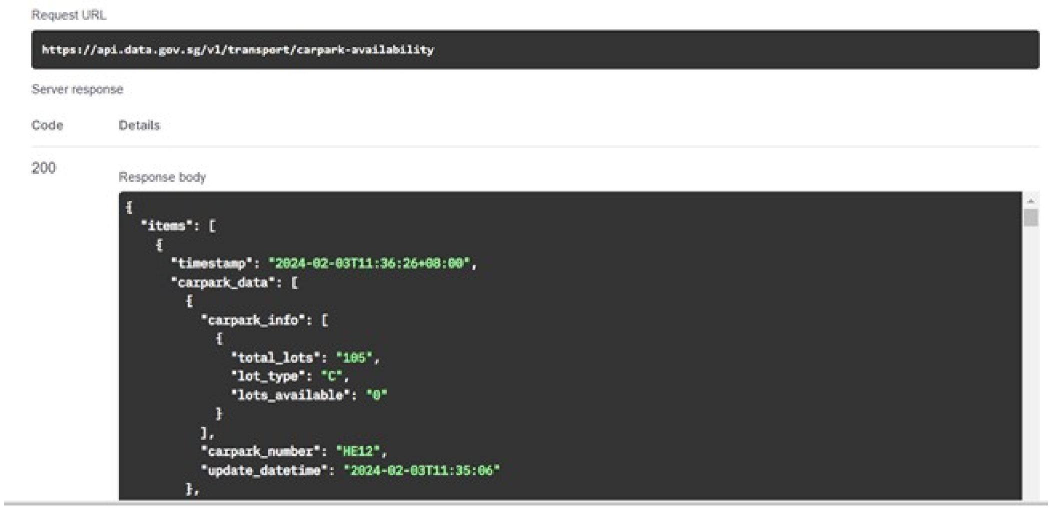
- Retrieved every minute
- Use the date_time parameter to retrieve the latest carpark availability at that moment in time
- Detailed carpark information can be found at https://data.gov.sg/dataset/hdb-carpark-information
- We recommend that this endpoint be called every minute

Parameters

Name	Description
date_time `string` *(query)*	YYYY-MM-DD[T]HH:mm:ss (SGT) 2023-12-25T:12:00:00

There are a few things to look out for in the output generated:

- Request URL: this is the URL that will be used in R to retrieve data.
- Code: this suggests if the request is successful or not. Commonly seen codes include 200 (successful), 400 (bad request), 404 (not found) and 501 (not implemented). Refer to https://umbraco.com/knowledge-base/http-status-codes/ for more information.
- Response body: this is the output that is generated based on the input parameters.


Request URL

```
https://api.data.gov.sg/v1/transport/carpark-availability
```

Server response

Code Details

200 Response body

```
{
  "items": [
    {
      "timestamp": "2024-02-03T11:36:26+08:00",
      "carpark_data": [
        {
          "carpark_info": [
            {
              "total_lots": "105",
              "lot_type": "C",
              "lots_available": "0"
            }
          ],
          "carpark_number": "HE12",
          "update_datetime": "2024-02-03T11:35:06"
        },
```

2. Within R, there are several ways to get data from API sources:

- General and preferred way is to use GET function from httr package

```r
library(httr)

api_url <- "https://api.data.gov.sg/v1/transport/carpark-availability"


given_datetime <- as.POSIXct("2023-12-25 12:00:00")
formatted_datetime <- format(given_datetime, format = "%Y-%m-%dT%H:%M:%S")


api_params <- list(
  date_time = formatted_datetime
  # other parameters...
)

response <- GET(api_url, query = api_params)
```

`response` is a list containing information about the request as well as the output. Print response to get some details such as date of request, status code, size and type of data.

```r
response
## Response [https://api.data.gov.sg/v1/transport/carpark-availability?date_time=2023-12-25T12%3A00%3A0
##   Date: 2024-02-04 07:15
##   Status: 200
##   Content-Type: application/json
##   Size: 296 kB
```

Depending on the data type, there are different ways of reading the output data into R. In this scenario, the output data is of json format. The json data can be retrieved in 2 ways as shown below (remember to install and import the "jsonlite" package first):

```r
library(jsonlite)

# first method
data <- fromJSON(rawToChar(response$content))

# second method
data <- fromJSON(content(response, "text"))
```

Finally, the dataframe object can be found by doing the following indexing:

```r
df <- data$items$carpark_data[[1]]

head(df)
##   carpark_info carpark_number     update_datetime
## 1    105, C, 0           HE12 2023-12-25T11:58:47
## 2  583, C, 474            HLM 2023-12-07T10:08:03
## 3  329, C, 149            RHM 2023-12-25T11:58:47
## 4    97, C, 28           BM29 2023-12-25T11:59:03
## 5    96, C, 31            Q81 2023-12-25T11:58:41
## 6  176, C, 133            C20 2023-12-07T05:00:44
```

- If the expected data format from the API is json, data can be directly retrieved in just a few lines of code with some knowledge of URL crafting.

The convention of crafting URL with multiple parameters is as follows:

<api_url>?<parameter1>=<value1>&<parameter2>=<value2>&...

In this example, this is the overall URL:

`https://api.data.gov.sg/v1/transport/carpark-availability?date_time=2023-12-25T12:00:00`

```r
library(jsonlite)

api_url <- "https://api.data.gov.sg/v1/transport/carpark-availability"

given_datetime <- as.POSIXct("2023-12-25 12:00:00")
formatted_datetime <- format(given_datetime, format = "%Y-%m-%dT%H:%M:%S")

data <- fromJSON(paste(api_url, "?date_time=", formatted_datetime,
    sep = ""))
```

Tips:

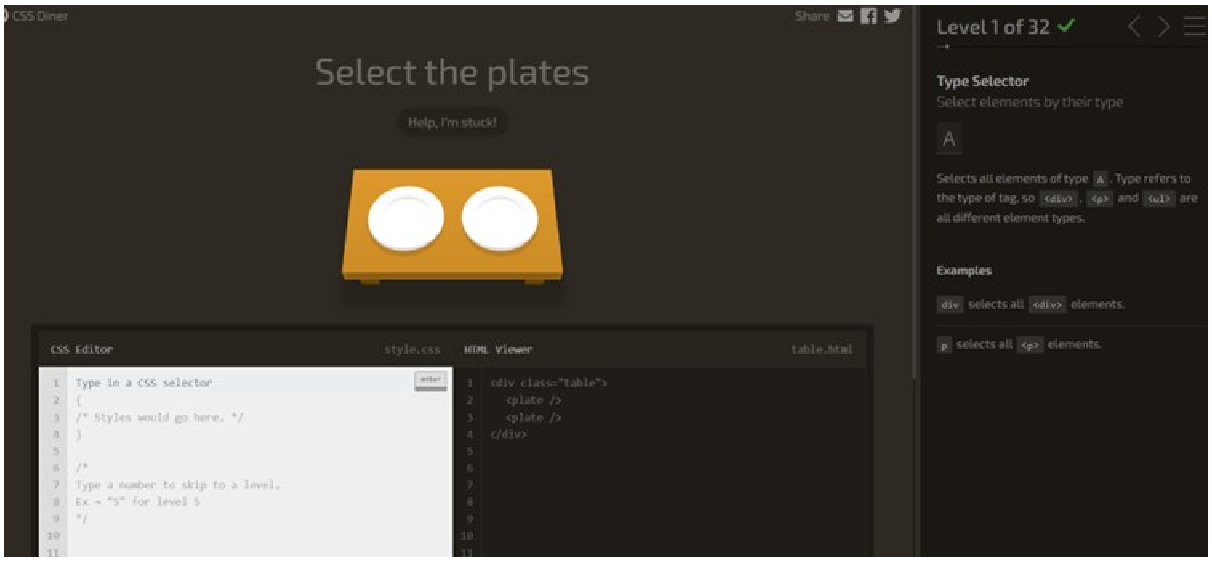
1. Due to the nature of json objects, it is a good practice to use `str(<json object>, max.level = <number>)` to get a better idea where the data is situated at. `max.level` argument is to specify how deep the `str()` function should print

Web Scraping

Websites can contain a lot of valuable information for analysis. However, not all can be obtained easily, e.g. via APIs, for instance, to get the latest news article links from a reputable news platform. Under such circumstances, web scraping is a valuable tool.

Finding the right elements in a website to extract can be a challenge without some background knowledge in html but there exist tools to make the process much easier. For a simple exercise on how we can make use of html nodes for web scraping, go to https://flukeout.github.io/#.

In this game, we are supposed to select elements of interests based on the instructions. For instance, in level 1, assuming that there is some information that is contained inside the plates (for now there is nothing) and we would like to extract it. We can extract that information by typing "plate" in the CSS editor and pressing Enter (See the instruction on the right), that would select all the plates that exist in the webpage. Hover over the plates to see the corresponding element in the HTML Viewer light up. Remember the things/keywords that you typed in the CSS editor, this will set precedence on what to expect in the web scraping example later.



```
Type in a CSS selector
{
/* Styles would go here. */
}

/*
Type a number to skip to a level.
Ex → "5" for level 5
*/
```

```
<div class="table">
  <plate />
  <plate />
</div>
```

Type Selector

Select elements by their type

A

Selects all elements of type `A`. Type refers to the type of tag, so `<div>`, `<p>` and `<ul>` are all different element types.

Examples

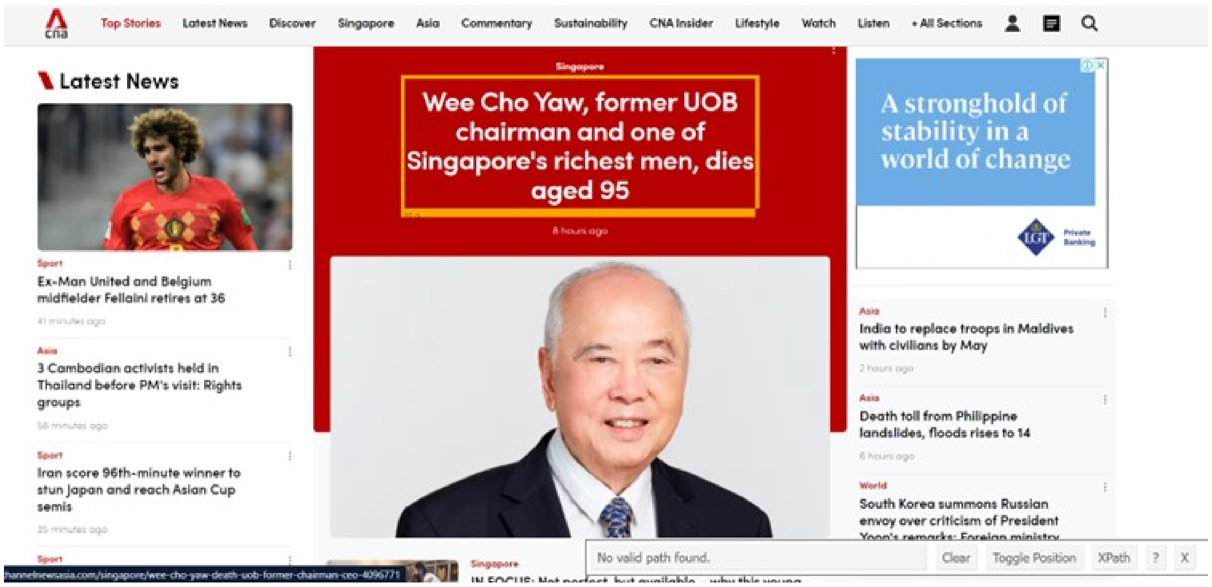
`div` selects all `<div>` elements.

`p` selects all `<p>` elements.

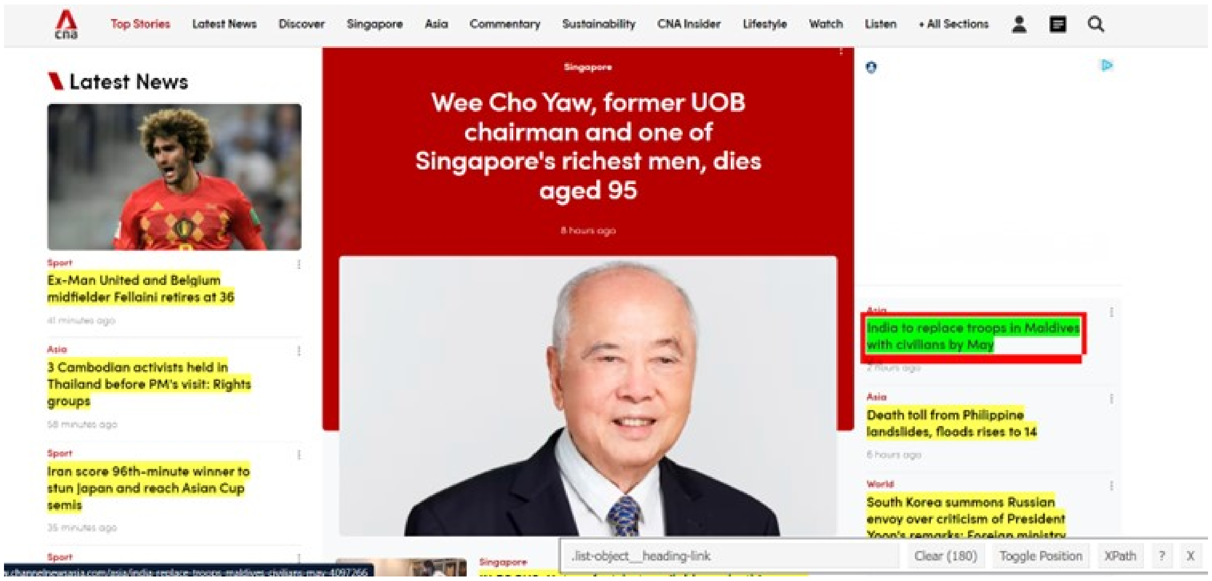
There are several ways to do web scraping. The harder way involves inspecting the elements (F12) of the website and finding the right keywords in order to extract the right elements. A much easier way is to use SelectorGadget which can be downloaded as an extension: https://chromewebstore.google.com/detail/selectorgadget/mhjhnkcfbdhnjickkkdbjoemdmbfginb. This tool provides us with the required keywords when we click on an element on the website.

In this example, we will be getting the names of articles found in Channel News Asia.

1. Click on the extensions tab and press SelectorGadget to activate it. To be sure that SelectorGadget is activated, notice the appearance of an orange box whenever you hover your cursor onto something on the webpage. Also there is a popup on the bottom right of the website.


Top Stories
Latest News
Discover
Singapore
Asia
Commentary
Sustainability
CNA Insider
Lifestyle
Watch
Listen
+ All Sections
Latest News
Sport
Ex-Man United and Belgium midfielder Fellaini retires at 36
41 minutes ago
Asia
3 Cambodian activists held in Thailand before PM's visit: Rights groups
58 minutes ago
Sport
Iran score 96th-minute winner to stun Japan and reach Asian Cup semis
35 minutes ago
Singapore
Wee Cho Yaw, former UOB chairman and one of Singapore's richest men, dies aged 95
8 hours ago
A stronghold of stability in a world of change
Asia
India to replace troops in Maldives with civilians by May
2 hours ago
Asia
Death toll from Philippine landslides, floods rises to 14
6 hours ago
World
South Korea summons Russian envoy over criticism of President Yoon's remarks: Foreign ministry
No valid path found.
Clear
Toggle Position
XPath
?
X

2. To get the titles of the articles, click on any of the titles. Notice how the clicked title appears green and other titles are highlighted in yellow. This shows all elements that are selected as well. Also notice the keyword that appears in the popup at the bottom. This is the keyword `.list-object__heading-link` required to select all the elements that are in green and yellow.


Top Stories
Latest News
Discover
Singapore
Asia
Commentary
Sustainability
CNA Insider
Lifestyle
Watch
Listen
+ All Sections
Latest News
Sport
Ex-Man United and Belgium midfielder Fellaini retires at 36
Asia
3 Cambodian activists held in Thailand before PM's visit: Rights groups
Sport
Iran score 96th-minute winner to stun Japan and reach Asian Cup semis
Singapore
Wee Cho Yaw, former UOB chairman and one of Singapore's richest men, dies aged 95
8 hours ago
Asia
India to replace troops in Maldives with civilians by May
Asia
Death toll from Philippine landslides, floods rises to 14
World
South Korea summons Russian envoy over criticism of President Yoon's remarks: Foreign ministry
.list-object__heading-link
Clear (180)
Toggle Position
XPath
?
X

3. Moving on to R, remember to install the package rvest, which contains functions to help us in web scraping. Execute the following code below. `html_nodes` function will extract all the relevant information that is contained within the html script based on the keyword provided. Print nodes to get an idea of what is being extracted.

Note: If `html_nodes` gives an error, install xml2 package and use the `html_nodes` function from there instead.

```r
library(rvest)

cna_page <- read_html("https://www.channelnewsasia.com/")

nodes <- html_nodes(cna_page, ".list-object__heading-link")
```

4. Depending on the complexity of the html code, there might be several extracted information within a block which can be hard to identify. Run the function `html_structure` to see what attributes are there within the node. The code below just focuses on the first node and the node only contains href which is the link to the news article.

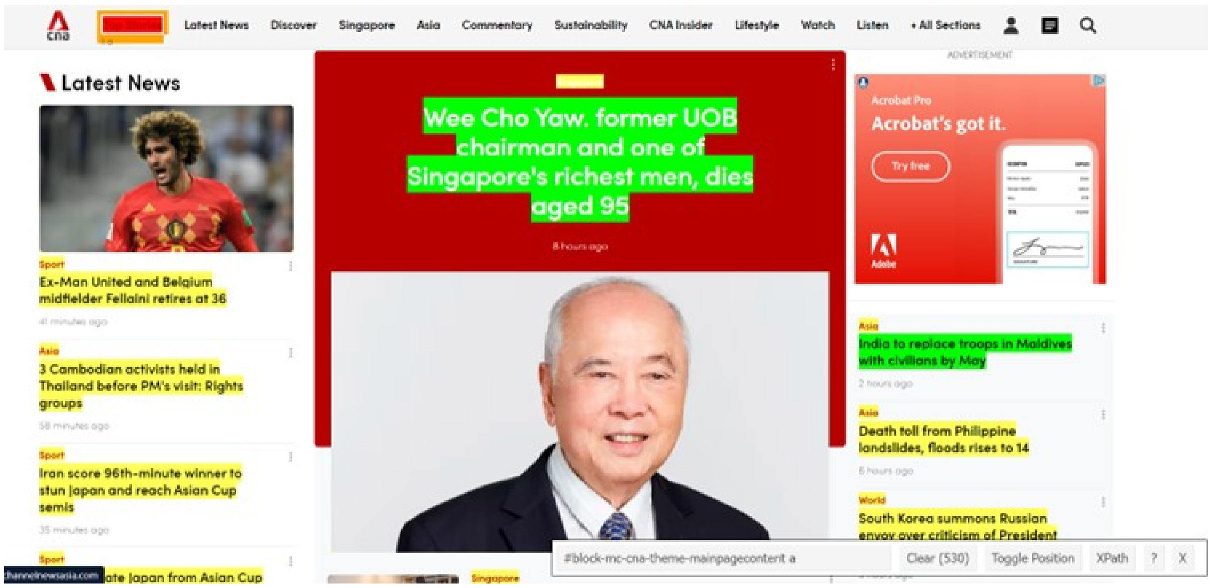
```r
html_structure(nodes[1])
## [[1]]
## <a.h6__link.list-object__heading-link [href]>
##   {text}
```

5. Run `html_attr` and give the attribute that you want to extract to get the right information. Further manipulation can be done to obtain the category and title of the article.

```r
head(html_attr(nodes, name = "href"))
## [1] "/singapore/lazada-retrenchment-fdawu-ntuc-package-benefits-4098186"
## [2] "/sport/howe-hopes-gordons-injury-not-serious-newcastles-woes-persist-4098336"
## [3] "/sport/india-lead-273-v-england-despite-flurry-wickets-4098306"
## [4] "/sport/ravindra-williamson-tons-frustrate-south-africa-new-zealand-4098291"
## [5] "/singapore/commonwealth-avenue-chopper-knife-attack-five-members-public-help-police-arrest-4098
## [6] "/sport/under-pressure-hodgson-says-he-can-turn-crystal-palace-around-4098191"
```

Tips:

1. Several elements can be selected by SelectorGadget and SelectorGadget could still output a single keyword in some cases. Hence, `html_nodes` only needs to run once.
2. Using SelectorGadget might involve some trial and error to get all the relevant information that the user needs to. For the above example, notice how the main article title in the middle is not highlighted. This is because the title requires a different keyword from the rest. In order to correctly capture that title without selecting too many unnecessary information, click on the title in the middle (see the green highlighted title) and then click the Top Stories header at the top. This will exclude the headers (see red highlighted) which was unintentionally included when clicking on the middle title.



Latest News

Sport
Ex-Man United and Belgium midfielder Fellaini retires at 36
41 minutes ago

Asia
3 Cambodian activists held in Thailand before PM's visit: Rights groups
58 minutes ago

Sport
Iran score 96th-minute winner to stun Japan and reach Asian Cup semis
35 minutes ago

Sport
ate Japan from Asian Cup

Wee Cho Yaw, former UOB chairman and one of Singapore's richest men, dies aged 95
8 hours ago

Singapore

Asia
India to replace troops in Maldives with civilians by May
2 hours ago

Asia
Death toll from Philippine landslides, floods rises to 14
6 hours ago

World
South Korea summons Russian envoy over criticism of President

channelnewsasia.com

```
#block-mc-cna-theme-mainpagecontent a    Clear (530)    Toggle Position    XPath    ?    X
```