



# OpenWeather APIs Calls

Estimated time needed: **40** minutes

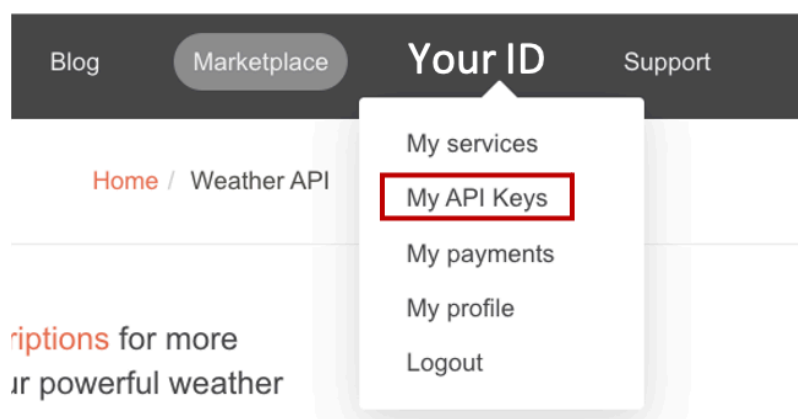
## Lab Overview:

In this lab, you will be collecting real-time current and forecasted weather data for cities using the **OpenWeather API**. It can give you current weather data for any location including over 200,000 cities and 5 day forecasts for free (with limited API usage). You just need to use HTTP requests to call those weather APIs and get the weather data.

You will be given instructions to set up a free OpenWeather API account with an API key for authentication. Then, you will be shown, with code examples, how to get the current weather for a given city. After you are familiar with OpenWeather API, you will be asked to get 5-day forecast data for a list of cities.

## Setup OpenWeather API account

You can create an OpenWeather API account [here](#). After your account has been created and verified, you can go to Account -> My API Keys:



and find your API key here:

Key	Name
eeadb0857cce348954f4217a0e6fd15c	sample

Now mark down your generated API key, which will be used to authenticate your HTTP requests to OpenWeather API.

Note that a new API key may take some time to be activated. You may quickly try the following URL using a web browser.

With the `{your_api_key}` URL parameter replaced by your own API key and go to the following URL:

[https://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=%7Byour\\_api\\_key%7D](https://api.openweathermap.org/data/2.5/weather?q=Seoul&appid=%7Byour_api_key%7D)

until you see some weather data returned (instead of 401 or other error status), similar to the following JSON result:

```
{
  "coord": {
    "lon": 126.9778,
    "lat": 37.5683
  },
  "weather": [
    {
      "id": 800,
      "main": "Clear",
      "description": "clear sky",
      "icon": "01n"
    }
  ],
  "base": "stations",
  "main": {
    "temp": 285.16,
    "feels_like": 284.04,
    "temp_min": 284.15,
    "temp_max": 287.15,
    "visibility": 10000,
    "wind": {
      "speed": 1.03,
      "deg": 220
    },
    "clouds": {
      "all": 0
    },
    "dt": 1617718307,
    "sys": {
      "type": 1,
      "id": 8105,
      "country": "KR",
      "sunrise": 1617657021,
      "sunset": 1617760000
    }
  }
}
```

## Coding Practice: Get the current weather data for a city using OpenWeather API

First import `httr` library. Run `install.packages("httr")` prior to loading the package only if you are running this lab locally on RStudio on your system.

```
In [ ]: # Check if need to install rvest` Library

library(httr)
```

The API base URL to get current weather is

<https://api.openweathermap.org/data/2.5/weather>

```
In [ ]: # URL for Current Weather API
current_weather_url <- 'https://api.openweathermap.org/data/2.5/weather'
```

Next, let's create a list to hold URL parameters for current weather API

```
In [ ]: # need to be replaced by your real API key
your_api_key <- "your_api_key"
# Input `q` is the city name
# Input `appid` is your API KEY,
# Input `units` are preferred units such as Metric or Imperial
current_query <- list(q = "Seoul", appid = your_api_key, units="metric")
```

Now we can make a HTTP request to the current weather API

```
In [ ]: response <- GET(current_weather_url, query=current_query)
```

If we check the response type, we can see it is in JSON format

```
In [ ]: http_type(response)
```

JSON is an open standard file and data interchange format that uses human-readable text to store and transmit data objects. To read the JSON HTTP response, you can use the `content()` function to parse it as a named list in R.

```
In [ ]: json_result <- content(response, as="parsed")
```

If you use the `class()` function, you can see it is a R `List` object

```
In [ ]: class(json_result)
```

Now let's print the JSON result.

```
In [ ]: json_result
```

It contains very detailed weather data about the city of `Seoul`. Feel free to try other cities as well. We need to convert the named list to a data frame so that we can use data frame operations to process the data. Below is a simple example, which you may implement your own way to convert it to a data frame.

```
In [ ]: # Create some empty vectors to hold data temporarily
weather <- c()
visibility <- c()
temp <- c()
temp_min <- c()
temp_max <- c()
pressure <- c()
humidity <- c()
wind_speed <- c()
wind_deg <- c()
```

Now assign the values in the `json_result` list into different vectors

```
In [ ]: # $weather is also a list with one element, its $main element indicates the weat
weather <- c(weather, json_result$weather[[1]]$main)
# Get Visibility
visibility <- c(visibility, json_result$visibility)
# Get current temperature
temp <- c(temp, json_result$main$temp)
# Get min temperature
temp_min <- c(temp_min, json_result$main$temp_min)
# Get max temperature
temp_max <- c(temp_max, json_result$main$temp_max)
# Get pressure
pressure <- c(pressure, json_result$main$pressure)
# Get humidity
humidity <- c(humidity, json_result$main$humidity)
```

```
# Get wind speed
wind_speed <- c(wind_speed, json_result$wind$speed)
# Get wind direction
wind_deg <- c(wind_deg, json_result$wind$deg)
```

Combine all vectors as columns of a data frame

```
In [ ]: # Combine all vectors
weather_data_frame <- data.frame(weather=weather,
                                visibility=visibility,
                                temp=temp,
                                temp_min=temp_min,
                                temp_max=temp_max,
                                pressure=pressure,
                                humidity=humidity,
                                wind_speed=wind_speed,
                                wind_deg=wind_deg)
```

```
In [ ]: # Check the generated data frame
print(weather_data_frame)
```

## TASK: Get 5-day weather forecasts for a list of cities using the OpenWeather API

Now you should be familiar with the usage of OpenWeather API. Next, you need to complete a task to get 5-day weather forecasts for a list of cities

*TODO:* Write a function to return a data frame containing 5-day weather forecasts for a list of cities

```
In [ ]: # Create some empty vectors to hold data temporarily

# City name column
city <- c()
# Weather column, rainy or cloudy, etc
weather <- c()
# Sky visibility column
visibility <- c()
# Current temperature column
temp <- c()
# Max temperature column
temp_min <- c()
# Min temperature column
temp_max <- c()
# Pressure column
pressure <- c()
# Humidity column
humidity <- c()
# Wind speed column
wind_speed <- c()
# Wind direction column
wind_deg <- c()
# Forecast timestamp
forecast_datetime <- c()
# Season column
```

```
# Note that for season, you can hard code a season value from Levels Spring, Summer, Autumn, Winter
season <- c()
```

```
In [ ]: # Get forecast data for a given city list
get_weather_forecast_by_cities <- function(city_names){
  df <- data.frame()
  for (city_name in city_names){
    # Forecast API URL
    forecast_url <- 'https://api.openweathermap.org/data/2.5/forecast'
    # Create query parameters
    forecast_query <- list(q = city_name, appid = "{your_api_key}", units="metric")
    # Make HTTP GET call for the given city

    # Note that the 5-day forecast JSON result is a list of lists. You can parse it with jsonlite
    #results <- json_list$list

    # Loop the json result
    for(result in results) {
      city <- c(city, city_name)

    }

    # Add the R Lists into a data frame
  }

  # Return a data frame
  return(df)
}
```

Complete and call `get_weather_forecast_by_cities` function with a list of cities, and write the data frame into a csv file called `cities_weather_forecast.csv`

```
In [ ]: cities <- c("Seoul", "Washington, D.C.", "Paris", "Suzhou")
cities_weather_df <- get_weather_forecast_by_cities(cities)
```

```
In [ ]: # Write cities_weather_df to `cities_weather_forecast.csv`
write.csv(cities_weather_df, "cities_weather_forecast.csv", row.names=FALSE)
```

For more details about HTTP requests with `httr`, please refer to the previous HTTP request notebook here:

[HTTP request in R](#)

## TASK: Download datasets as csv files from cloud storage

The last task of this lab is straightforward: download some aggregated datasets from cloud storage

```
In [ ]: # Download several datasets

# Download some general city information such as name and locations
url <- "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDe
# download the file
```

```
download.file(url, destfile = "raw_worldcities.csv")

# Download a specific hourly Seoul bike sharing demand dataset
url <- "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDa
# download the file
download.file(url, destfile = "raw_seoul_bike_sharing.csv")
```

## Next Step

Now you should collect all datasets required for this project, all in csv format. Feel free to explore them preliminarily now. However, they are the raw datasets and have to be processed for further data visualization or data analysis.

In the next module, you will improve their quality by performing data wrangling tasks.

## Authors

[Yan Luo](#)

## Other Contributors

© IBM Corporation 2021. All rights reserved.

In [ ]: