

Lecture 10: API and R

Accessing Accuweather Automatically

Isabel Casas
icasas@deusto.es

What is an API?

- It is an interface between us and a server.
- We can use it to access to data from servers like twitter, google, idealista, etc.
- Companies have documentation about their API to help developers to use the data they are happy to share:
 - ▶ Accuweather
 - ▶ Twitter

How to access the API of a company using R

- Get invited: Get a developer account, create an app to get an API key and OAuth authorisation
- Access their API **GET** and/or **POST** functions using the **GET** and/or **POST** functions in R package **httr**
- Convert the data given in something readable and usable in R by using R packages **jsonlite** or **xml2**
 - ▶ What is json
 - ▶ What is XML

How to access the API of a company using R

- There are specialised packages in R for certain websites (like for Twitter). For example,
- Rfacebook
- rtwit, twitterR
- instaR
- Many more
- For the rest, we have to program it ourselves using the *httr* package like we are seeing here.

Example: accessing the Accuweather API

A video in: [How to get Accuweather API key](#) explains how to use the **Accuweather API** in R.

Steps:

- 1 Check the API REFERENCE to see what kind of data Accuweather allows us to collect. The place to start: [ACCUWEATHER website](#).
- 2 Create a developer account and an application. This will generate an API key, which is a number like Oqf08xM5gTIVtzFbzkwSTL2w2V9. We will need this API key in our queries.
- 3 Check the format API functions available in the API REFERENCE to choose what you want to request from that website.
 - ▶ We are going to use the **Location API/City Search** to find the Accuweather code for Donosti
 - ▶ Then we are going to use the **Forecast API/Forecast 1 Day of Daily Forecasts** to get the prediction for Donosti's weather tomorrow
- 4 Make requests using functions GET and POST

First: create the URL from where to make a request

- We need to check the GET API function on the website. For example, to check the location code of a city we see in the **Location API/City Search** the following:

Resource URL

`http://dataservice.accuweather.com/locations/v1/cities/search`

Query Parameters

Name	Values	Description
apikey (required)	<input type="text"/>	Provided API Key.
q (required)	<input type="text"/>	Text to search for.
language	<input type="text" value="en-us"/>	String indicating the language in which to return the resource.
details	<input type="text" value="false"/>	Boolean value specifies whether or not to include full details in the response.
offset	<input type="text"/>	Integer, along with the limit (25) that determines the first resource to be returned. If no offset is provided, the max number (100) of results will be returned.
alias	<input type="text" value="NoOfficialMatchFound"/>	Enumeration that specifies when alias locations should be included in the results. By default, an alias will only be returned if no official match for the search text was found. Enumeration values: Never or Always

First: create the URL from where to make a request

- We see that the “end-point” or “resource URL” is:
<http://dataservice.accuweather.com/locations/v1/cities/search>
- That query has 2 required parameters: *apikey* and *q* that must be included in our URL.
- So we will create the following URL:

[http://dataservice.accuweather.com/locations/v1/cities/
search?apikey=YOURAPI&q=YOURCITY](http://dataservice.accuweather.com/locations/v1/cities/search?apikey=YOURAPI&q=YOURCITY)

First: create the URL from where to make a request

```
library(httr)
library(jsonlite)
library(dplyr)
Sys.setenv(ACCUWEATHER_KEY = "YOURKEY") # save your akey
city="Bilbao"
myurl1 <- paste0("http://dataservice.accuweather.com/locations/v1/cities/autocomplete?apikey=",
                 Sys.getenv("ACCUWEATHER_KEY"), "&q=", city)
accweather_raw1 <- httr::GET(myurl1)
names(accweather_raw1)
cat("The status code is: ", accweather_raw1$status_code)
```

Function `paste0` glue together text and variables in a big string and we use it to generate this URL.

```
myurl1 <- paste0("http://dataservice.accuweather.com/locations/v1/cities/autocomplete?apikey=",
                 Sys.getenv("ACCUWEATHER_KEY"), "&q=", city)
```


Output of our R chunk

```
## [1] "url"          "status_code" "headers"      "all_headers" "cookies"
## [6] "content"      "date"         "times"        "request"      "handle"
## The status code is: 200
```

Output of our R chunk

We see our request has a status code of 200, which means it has been able to make contact, other codes are:

Response Error Details

HTTP Code	Error Code	Description
400		Request had bad syntax or the parameters supplied were invalid.
401		Unauthorized. API authorization failed.
403		Unauthorized. You do not have permission to access this endpoint.
404		Server has not found a route matching the given URI.
500		Server encountered an unexpected condition which prevented it from fulfilling the request.

It also has other information like the content that is of our interest, but not very readable when it is raw:

```
head(accweather_raw1$content)
```

```
## [1] 5b 7b 22 56 65 72
```

Second: read the content

That is why we need to convert it into text using function *content* from package *httr*.

```
accweather_content<- httr::content(accweather_raw1, as = "text")  
str(accweather_content)
```

```
## chr "[{"Version":1,"Key":"309382","Type":"City","Rank":32,"LocalizedName":"Bilbao","Country":...}]"
```

Second: read the content

Even though the result is data in a JSON format in this case, that will need to convert into a data.frame or matrix in R to be able to read it.

```
mycontent <- jsonlite::fromJSON(accweather_content)
mycontent
```

##	Version	Key	Type	Rank	LocalizedName	Country.ID	Country.Localized
## 1	1	309382	City	32	Bilbao	ES	S
## 2	1	106790	City	85	Bilbao	CO	Colo
## 3	1	2330252	City	85	Bilbao	ES	S
## 4	1	306186	City	85	Bilbao La Vieja	ES	S
## 5	1	780840	City	85	Bilbao	PH	Philipp
##	AdministrativeArea.ID		AdministrativeArea.LocalizedName				
## 1			PV		Basque Country		
## 2			TOL		Tolima		
## 3			MD		Madrid		
## 4			PV		Basque Country		
## 5			ALB		Albay		

Create the URL and do your request

- Now my content is a *data.frame*, something that R understands
- There are a few cities in the world with the name Bilbao. Our *Key* = 309382.
- We can do it manually or automatically:

```
city.key <- mycontent$Key[mycontent$AdministrativeArea[["LocalizedName"]]=="Basque Country"]  
city.key
```

```
## [1] "309382" "306186"  
mycity.key <- city.key[1]
```

Activity 1

Find out the city key in the Accuweather database of Vitoria, Durango and San Sebastian.

Example: collect 1 day ahead forecast weather

- 1 Go to the **Forecast API** and create the URL to obtain the 1 day ahead forecast of weather in Bilbao.
- 2 Read the content and convert it into something R can use
- 3 Then obtain the value

Example: collect 1 day ahead forecast weather

In the API REFERENCE, we get the URL source and its parameters.

```
myurl2 <- paste0("http://dataservice.accuweather.com/forecasts/v1/daily/1day/",  
                mycity.key, "?apikey=", Sys.getenv("ACCUWEATHER_KEY"))  
accweather_raw2 <- httr::GET(myurl2)  
accweather_content <- httr::content(accweather_raw2, as = "text")  
for.1d <- jsonlite::fromJSON(accweather_content)  
names(for.1d)
```

```
## [1] "Headline"      "DailyForecasts"
```

```
class(for.1d)
```

```
## [1] "list"
```


Example: collect 1 day ahead forecast weather

- **for.1d** is an object of class *list* and therefore we have to use `[[]]` to access its 2 components: *Headline* and *DailyForecasts*.
- Not every city has the same number of values because it depends on the meteorological stations.

Example: collect 1 day ahead forecast weather

To obtain the 1-day weather forecast for Donosti tomorrow:

```
forecast <- for.1d[["DailyForecasts"]]  
forecast$Temperature$Minimum$Value
```

```
## [1] 44
```

```
forecast$Temperature$Maximum$Value
```

```
## [1] 65
```

```
forecast$Day$PrecipitationType
```

```
## [1] "Rain"
```

```
forecast$Day$PrecipitationIntensity
```

```
## [1] "Light"
```

The temperature is not in degree Celsius, we can see that in:

```
forecast$Temperature$Minimum$Unit
```

```
## [1] "F"
```

How to do this for a list of cities

- First, we have to obtain the code for all the cities of interest. Then do a look over all.
- For example, we want to obtain results for Sydney, Boston, London, Bilbao and Rabat.

```
city.list <- c("Sydney", "Boston", "London", "Bilbao", "Rabat")
country <- c("Australia", "United States", "United Kingdom", "Spain", "Morocco")
city.key <- numeric(5)
for (i in 1:5)
{
  myurl1 <- paste0("http://dataservice.accuweather.com/locations/v1/cities/autocomplete?apikey=",
                  Sys.getenv("ACCUWEATHER_KEY"), "&q=", city.list[i])
  raw <- httr::GET(myurl1)
  if(raw$status_code >= 400)
    stop("Error in connection with city ", city.list[i])
  content<- httr::content(raw, as = "text")
  mycontent <- jsonlite::fromJSON(content)
  city.key[i] <- mycontent$key[mycontent$Country$LocalizedName==country[i]]
}
city.key
```

```
## [1] "22889" "348735" "328328" "309382" "245072"
```

How to do this for a list of cities

Now we have all the city codes, we will find the 1 day ahead forecast for each and make a table with *city name*, *code*, *minimum* and *maximum* temperature as well as the *weather prediction*

```
mymatrix <- data.frame(City = city.list, Code = city.key)
min = max= weather<- numeric(5)
for (i in 1:5)
{
  myurl1 <- paste0("http://dataservice.accuweather.com/forecasts/v1/daily/1day/",
                  city.key[i], "?apikey=",
                  Sys.getenv("ACCUWEATHER_KEY"))
  raw <- httr::GET(myurl1)
  if(raw$status_code >=400)
    stop("Error in connection with city ", city.list[i])
  content<- httr::content(raw, as = "text")
  mycontent <- jsonlite::fromJSON(content)
  forecast <- mycontent[["DailyForecasts"]]
  min[i] <- forecast$Temperature$Minimum$Value
  max[i] <- forecast$Temperature$Maximum$Value
  weather[i] <- forecast$Day$IconPhrase
}
mymatrix <- data.frame(mymatrix, MinimumTemp = min, MaximumTemp=max, Weather= weather)
mymatrix
```

##	City	Code	MinimumTemp	MaximumTemp	Weather
## 1	Sydney	22889	64	79	Mostly cloudy
## 2	Boston	348735	50	71	Intermittent clouds
## 3	London	328328	47	61	Partly sunny
## 4	Bilbao	309382	44	65	Showers
## 5	Rabat	245072	49	64	Partly sunny w/ showers

Activity 2

Find another website with an API service?