

Assignment9 - Web APIs

Jawaid Hakim

2022-10-28

Contents

1	Load libraries	1
2	New York Time Books API Vignette	2
3	Create R API	2
4	API Test	7
4.1	Names of all Best Seller lists	7
4.2	Best Seller list	7
4.3	Best Seller list by date	8
4.4	Best Seller list history	9
4.4.1	By age group	9
4.4.2	By age group and publisher	10
4.4.3	By publisher and price	10
4.4.4	By publisher and offset (offset must be a multiple of 20)	11
4.5	All books for all the Best Sellers lists for specified date	12

1 Load libraries

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr  0.3.5
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(jsonlite)
```

```
##  
## Attaching package: 'jsonlite'  
##  
## The following object is masked from 'package:purrr':  
##  
##   flatten
```

```
library(assertive) # for validating request parameters
```

```
##  
## Attaching package: 'assertive'  
##  
## The following objects are masked from 'package:purrr':  
##  
##   is_atomic, is_character, is_double, is_empty, is_formula,  
##   is_function, is_integer, is_list, is_logical, is_null, is_numeric,  
##   is_vector  
##  
## The following object is masked from 'package:tibble':  
##  
##   has_rownames
```

2 New York Time Books API Vignette

In this solution we will use the New York Times [Books API](#) to retrieve information about The New York Times Best Sellers lists.

3 Create R API

Set the API Key in one place to make it easy to change.

```
API_KEY <- "2mGra8XCfEhUdE4WxThgvyCs55BZLfLK"
```

Create utilities to facilitate development of API functions. These include URL encoding (to escape reserved characters), modifying URL placeholders, adding request parameters, and adding API access key.

```
## Finalize URL. Add API access key top URL and URLEncode.  
##  
## @param url URL  
## @param verbose - TRUE to print final URL.  
## @returns URL with API access key and URL encoding  
FinalizeUrl <- function(url, verbose = FALSE) {  
  if (str_detect(url, "\\?")) {  
    url <- paste0(url, "&")  
  } else {  
    url <- paste0(url, "?")  
  }  
}
```

```

}
url <- paste0(url, "api-key=", API_KEY)
url <- URLEncode(url)
if (verbose) {
  print(url)
}
return (url)
}

#' URL component substitution and request parameter injection.
#'
#' @param url URL
#' @param url_subs URL substitution dictionary. Keys are placeholders in the URL and values are substitutions.
#' @param req_params Request parameter dictionary. Keys are parameter names and values are parameter substitutions.
#' @returns URL with URL substitutions and request parameters.
#' @example AddReqParams("http://foo.com/{book}/index.json", c('{book}':'fiction'),c('param':'123'))
AddReqParams <- function(url, url_subs = NULL, req_params = NULL) {
  if (!is_empty(url_subs)) {
    for (key in names(url_subs)) {
      url <- str_replace(url, key, url_subs[key])
    }
  }

  if (!is_empty(req_params)) {
    first <- TRUE
    for (key in names(req_params)) {
      if (first) {
        url <- paste0(url, "?")
        first <- FALSE
      } else {
        url <- paste0(url, "&")
      }
      url <- paste0(url, key, "=", req_params[key])
    }
  }
  return (url)
}

#' URL component substitution, request parameter injection, API key injection, and URL encoding.
#'
#' @param url URL
#' @param url_subs URL substitution dictionary. Keys are placeholders in the URL and values are substitutions.
#' @param req_params Request parameter dictionary. Keys are parameter names and values are parameter substitutions.
#' @returns URL with URL substitutions and request parameters.
#' @example AddReqParamsAndFinalize("http://foo.com/{book}/index.json", c('{book}':'fiction'),c('key':'123'))
AddReqParamsAndFinalize <- function(url, url_subs = NULL, req_params = NULL) {
  url <- AddReqParams(url, url_subs, req_params)
  url <- FinalizeUrl(url)
  return (url)
}

```

API endpoints.

```

BOOKS_API_VERSION <- "v3"

BOOKS_API_BASE_URL <- paste0("https://api.nytimes.com/svc/books", "/", BOOKS_API_VERSION)

BOOKS_API_LIST_NAMES_URL <- paste0(BOOKS_API_BASE_URL, "/lists/names.json")

BOOKS_API_LIST_URL <- paste0(BOOKS_API_BASE_URL, "/lists.json")

BOOKS_API_LIST_BYDATE_URL <- paste0(BOOKS_API_BASE_URL, "/lists/{date}/{list}.json")

BOOKS_API_HISTORY_URL <- paste0(BOOKS_API_BASE_URL, "/lists/best-sellers/history.json")

BOOKS_API_OVERVIEW_URL <- paste0(BOOKS_API_BASE_URL, "/lists/overview.json")

BOOKS_API_FULL_OVERVIEW_URL <- paste0(BOOKS_API_BASE_URL, "/lists/full-overview.json")

```

API functions. Each function encapsulates one New York Times Books API and the required and optional parameters.

```

#' Get Best Seller List names.
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists/names.json/get
API.GetNames <- function(verbose = FALSE) {
  url <- FinalizeUrl(BOOKS_API_LIST_NAMES_URL, verbose)
  o <- fromJSON(url)
  return (o)
}

#' Get Best Seller List.
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists.json/get
API.GetNamedList <- function(list_name, bestsellers_date = NULL, published_date = NULL, offset = NULL, verbose = FALSE) {
  url <- BOOKS_API_LIST_URL

  params <- c("list" = list_name)

  if (!is_empty(bestsellers_date)) {
    params["bestsellers-date"] <- bestsellers_date
  }

  if (!is_empty(published_date)) {
    params["published-date"] <- published_date
  }

  if (!is_empty(offset)) {
    assert_all_are_divisible_by(offset, 20) # must be multiple of 20
    params["offset"] <- str(offset)
  }

  url <- AddReqParams(url, NULL, params)
  url <- FinalizeUrl(url, verbose)
  o <- fromJSON(url)
  return (o)
}

```

```

}

#' Get Best Seller list
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists/%7Bdate%7D/%7Blist%7D.json/get
API.GetListByDate <- function(list_date = "current", list_name, offset = NULL, verbose = FALSE) {
  url <- BOOKS_API_LIST_BYDATE_URL

  url_subs <- c("\\{date\\}" = list_date, "\\{list\\}" = list_name)

  params <- c()
  if (!is_empty(offset)) {
    assert_all_are_divisible_by(offset, 20) # must be multiple of 20
    params["offset"] <- str(offset)
  }

  url <- AddReqParams(url, url_subs, params)
  url <- FinalizeUrl(url, verbose)
  o <- fromJSON(url)
  return (o)
}

#' Get Best Seller list history.
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists/best-sellers/history.json/get
API.GetHistory <- function(age_group = NULL,
                           author = NULL,
                           contributor = NULL,
                           isbn = NULL,
                           offset = NULL,
                           price = NULL,
                           publisher = NULL,
                           title = NULL,
                           verbose = FALSE) {
  url <- BOOKS_API_HISTORY_URL

  params <- c()

  if (!is_empty(age_group)) {
    params["age-group"] <- age_group
  }

  if (!is_empty(author)) {
    params["author"] <- author
  }

  if (!is_empty(contributor)) {
    params["contributor"] <- contributor
  }

  if (!is_empty(isbn)) {
    params["isbn"] <- isbn
  }
}

```

```

if (!is_empty(offset)) {
  assert_all_are_divisible_by(offset, 20) # must be multiple of 20
  params["offset"] <- offset
}

if (!is_empty(price)) {
  params["price"] <- price
}

if (!is_empty(age_group)) {
  params["age-group"] <- age_group
}

if (!is_empty(publisher)) {
  params["publisher"] <- publisher
}

if (!is_empty(title)) {
  params["title"] <- title
}

url <- AddReqParams(url, NULL, params)
url <- FinalizeUrl(url, verbose)
o <- fromJSON(url)
return (o)
}

#' Get all books for all the Best Sellers lists for specified date.
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists/full-overview.json/get
API.GetFullOverview <- function(published_date = NULL, verbose = FALSE) {
  url <- BOOKS_API_FULL_OVERVIEW_URL

  params <- c()
  if (!is_empty(published_date)) {
    params["published_date"] <- published_date
  }

  url <- AddReqParams(url, NULL, params)
  url <- FinalizeUrl(url, verbose)
  o <- fromJSON(url)
  return (o)
}

#' Get all books for all the Best Sellers lists for specified date.
#'
#' https://developer.nytimes.com/docs/books-product/1/routes/lists/overview.json/get
API.GetOverview <- function(published_date = NULL, verbose = FALSE) {
  url <- BOOKS_API_OVERVIEW_URL

  params <- c()
  if (!is_empty(published_date)) {
    params["published_date"] <- published_date
  }

```

```

}

url <- AddReqParams(url, NULL, params)
url <- FinalizeUrl(url, verbose)
o <- fromJSON(url)
return (o)
}

```

4 API Test

Time to put the APIs through their paces!

4.1 Names of all Best Seller lists

```
ret <- API.GetNames(verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/names.json?api-key=2mGra8XCfEhUdE4WxThgvyCs55BZLfLK"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results, n = 2)
}
```

```
##               list_name               display_name
## 1 Combined Print and E-Book Fiction Combined Print & E-Book Fiction
## 2 Combined Print and E-Book Nonfiction Combined Print & E-Book Nonfiction
##               list_name_encoded oldest_published_date
## 1 combined-print-and-e-book-fiction          2011-02-13
## 2 combined-print-and-e-book-nonfiction        2011-02-13
## newest_published_date updated
## 1          2022-11-06 WEEKLY
## 2          2022-11-06 WEEKLY
```

4.2 Best Seller list

Get list Advice How-To and Miscellaneous. List name has features (whitespace, -) that test URL encoding.

```
ret <- API.GetNamedList(list_name = 'Advice How-To and Miscellaneous', verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists.json?list=Advice%20How-To%20and%20Miscellaneous&api-key=2mGra8XCfEhUdE4WxThgvyCs55BZLfLK"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results, n = 2)
}
```

```
##               list_name               display_name
## 1 Advice How-To and Miscellaneous Advice, How-To & Miscellaneous
## 2 Advice How-To and Miscellaneous Advice, How-To & Miscellaneous
##   bestsellers_date published_date rank rank_last_week weeks_on_list asterisk
## 1      2022-10-22      2022-11-06   1             1         152         0
## 2      2022-10-22      2022-11-06   2             0          1         0
##   dagger
## 1      1
## 2      1
##
##               amazon_product_url
## 1 https://www.amazon.com/Atomic-Habits-Proven-Build-Break/dp/0735211299?tag=NYTBSREV-20
## 2      https://www.amazon.com/dp/0358539145?tag=NYTBSREV-20
##
## 1 0735211299, 0735211302, 1847941834, 0593207092, 1804225789, 9780735211292, 9780735211308, 97818479
## 2
##
##               book_details
## 1      ATOMIC HABITS, , by James Clear, James Clear, , 0.00, , Avery, 9780735211292, 0735211299
## 2 FEED THESE PEOPLE, , by Jen Hatmaker, Jen Hatmaker, , 0.00, , Harvest, 9780358539148, 0358539145
##   reviews
## 1      , , ,
## 2      , , ,
```

4.3 Best Seller list by date

```
ret <- API.GetListByDate(list_name = 'education', verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/current/education.json?api-key=2mGra8XCfEhUdE4WxThgv"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results$books, n = 2)
}
```

```
##   rank rank_last_week weeks_on_list asterisk dagger primary_isbn10
## 1    1              0              0      0      0      0374533555
## 2    2              0              0      0      0      1501111108
##   primary_isbn13      publisher
## 1 9780374533557 Farrar, Straus & Giroux
## 2 9781501111105      Scribner
##
## 1 A winner of the Nobel in economic science discusses how we make choices in business and personal l
## 2                                     A psychologist s
##   price      title      author      contributor
## 1 0.00 THINKING, FAST AND SLOW Daniel Kahneman by Daniel Kahneman
## 2 0.00      GRIT Angela Duckworth by Angela Duckworth
##   contributor_note
## 1
## 2
##
##               book_image
## 1 https://storage.googleapis.com/du-prd/books/images/9781429969352.jpg
## 2 https://storage.googleapis.com/du-prd/books/images/9781501111129.jpg
```



```
##      book_image_width book_image_height
## 1              330              490
## 2              328              495
##
##                                     amazon_product_url
## 1  http://www.amazon.com/Thinking-Fast-Slow-Daniel-Kahneman-ebook/dp/B00555X80A?tag=NYTBSREV-20
## 2  http://www.amazon.com/Grit-The-Power-Passion-Perseverance-ebook/dp/B010MH9V3W?tag=NYTBSREV-20
##      age_group
## 1
## 2
##
##                                     book_rev:
## 1  https://www.nytimes.com/2011/11/27/books/review/thinking-fast-and-slow-by-daniel-kahneman-book-rev:
## 2                                     https://www.nytimes.com/2016/05/08/books/review/grit-by-angela-duckwo:
##      first_chapter_link
## 1
## 2
##
##                                     sunday_rev:
## 1  https://www.nytimes.com/2011/11/27/books/review/thinking-fast-and-slow-by-daniel-kahneman-book-rev:
## 2
##      article_chapter_link
## 1
## 2
##
## 1                                     0374275637, 0739357980, 0374533555, 06062756
## 2  1501111108, 1501111124, 1442397144, 1501111116, 1442397136, 1501144162, 9781501111105, 97815011111
##
## 1  Amazon, Apple Books, Barnes and Noble, Books-A-Million, Bookshop, IndieBound, http://www.amazon.com
## 2
##
##                                     book_uri
## 1  nyt://book/1e1ffccd-d526-5154-84de-1628b31454e9
## 2  nyt://book/1e121a80-5501-5e47-be12-9169b765da78
```

4.4 Best Seller list history

4.4.1 By age group

```
ret <- API.GetHistory(age_group = 15, verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/best-sellers/history.json?age-group=15&api-key=2mGra"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results, n = 2)
}
```

```
##                                     title
## 1 ETIQUETTE AND ESPIONAGE
## 2      I HUNT KILLERS
##
##                                     description
## 1                                     Sophronia is sent to an unusual finishing school.
## 2 A teenager tries to control his own destiny in the face of overwhelming odds.
##      contributor      author contributor_note price      age_group
```

```
## 1 by Gail Carriger Gail Carriger 0.00 Ages 11 to 15
## 2 by Barry Lyga Barry Lyga 0.00 Ages 15 to 18
## publisher isbn
## 1 Little, Brown & Company 031619008X, 9780316190084
## 2 Little, Brown & Company 031620174X, 0316125849, 9780316201742, 9780316125840
## ranks_history
## 1 031619008X, 9780316190084, 9, Young Adult, Young Adult, 2013-02-24, 2013-02-09, 1, 0, 0, 0
## 2 031620174X, 9780316201742, 3, Young Adult, Young Adult, 2013-03-24, 2013-03-09, 1, 0, 0, 0
## reviews
## 1 , , ,
## 2 , , ,
```

4.4.2 By age group and publisher

```
ret <- API.GetHistory(age_group = 15, publisher = 'Bloomsbury', verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/best-sellers/history.json?age-group=15&publisher=Bloomsbury"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results)
}
```

```
## title
## 1 PRINCESS ACADEMY: PALACE OF STONE
## description
## 1 In this book that picks up where "Princess Academy" left off, a mountain girl moves to the city.
## contributor author contributor_note price age_group publisher
## 1 by Shannon Hale Shannon Hale 16.99 Ages 10 to 15 Bloomsbury
## isbn
## 1 1599908735, 9781599908731
## ranks_history
## 1 1599908735, 9781599908731, 8, Chapter Books, Children's Chapter Books, 2012-09-09, 2012-08-25, 1, 0, 0, 0
## reviews
## 1 , , ,
```

4.4.3 By publisher and price

```
ret <- API.GetHistory(publisher = 'HarperCollins', verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/best-sellers/history.json?publisher=HarperCollins&ap"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results, n = 2)
}
```

```
##           title
## 1          #ASKGARYVEE
## 2 100 DAYS OF REAL FOOD
##
## 1 The entrepreneur expands on subjects addressed on his Internet show, like marketing, management and
## 2                               Wholesome, family-friendly recipes that are ea
##           contributor      author contributor_note price age_group
## 1 by Gary Vaynerchuk Gary Vaynerchuk          0.00
## 2   by Lisa Leake      Lisa Leake          0.00
##           publisher                                     isbn
## 1      HarperCollins 0062273124, 0062273132, 9780062273123, 9780062273130
## 2 Morrow/HarperCollins 0062252550, 0062324098, 9780062252555, 9780062324092
##
## 1
## 2 0062252550, 0062252550, 0062252550, 0062252550, 0062252550, 9780062252555, 9780062252555, 97800622
##  reviews
## 1   , , ,
## 2   , , ,
```

4.4.4 By publisher and offset (offset must be a multiple of 20)

```
ret <- API.GetHistory(publisher = 'HarperCollins', offset = 20, verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/best-sellers/history.json?offset=20&publisher=Harper"
```

```
if (ret$status == "OK") {
  #str(ret)
  head(ret$results, n = 2)
}
```

```
##           title
## 1      A TIME FOR TRUTH
## 2 A VERY MARLEY CHRISTMAS
##
## 1 The Texas senator and Republican presidential candidate tells his personal and political story, and
## 2
##           contributor      author
## 1           by Ted Cruz      Ted Cruz
## 2 by John Grogan. Illustrated by Richard Cowdrey John Grogan
##           contributor_note price   age_group      publisher
## 1           0.00          Broadside/HarperCollins
## 2 Illustrated by Richard Cowdrey 17.99 Ages 3 to 8      HarperCollins
##           isbn
## 1 0062365630, 0062365614, 9780062365637, 9780062365613
## 2          NULL
##
## 1 0062365614, 0062365614, 0062365614, 0062365614, 0062365630, 0062365614, 0062365614, 0062365614, 00
## 2
##  reviews
## 1   , , ,
## 2   , , ,
```

4.5 All books for all the Best Sellers lists for specified date

```
ret <- API.GetOverview(published_date = '2013-05-15', verbose = TRUE)
```

```
## [1] "https://api.nytimes.com/svc/books/v3/lists/overview.json?published_date=2013-05-15&api-key=2mGr"
```

```
if (ret$status == "OK") {  
  print(ret$num_results)  
}
```

```
## [1] 105
```