

APIs & JSON

INTERMEDIATE IMPORTING DATA IN R



Filip Schouwenaars
Instructor, DataCamp

Other data formats

- Before: pages and files from the web
- JSON
- Simple, concise, well-structured
- Human-readable
- Easy to parse and generate for computers
- For communication with Web APIs

API

- Application Programming Interface
- Set of routines and protocols for building software
- How different components interact
- Web API
 - interface to get or add data to server
 - HTTP verbs (GET and others)

Twitter

- <https://dev.twitter.com/rest/public>
- Get tweets
- Place comments on tweets
- Many applications
 - Research effect of tweets

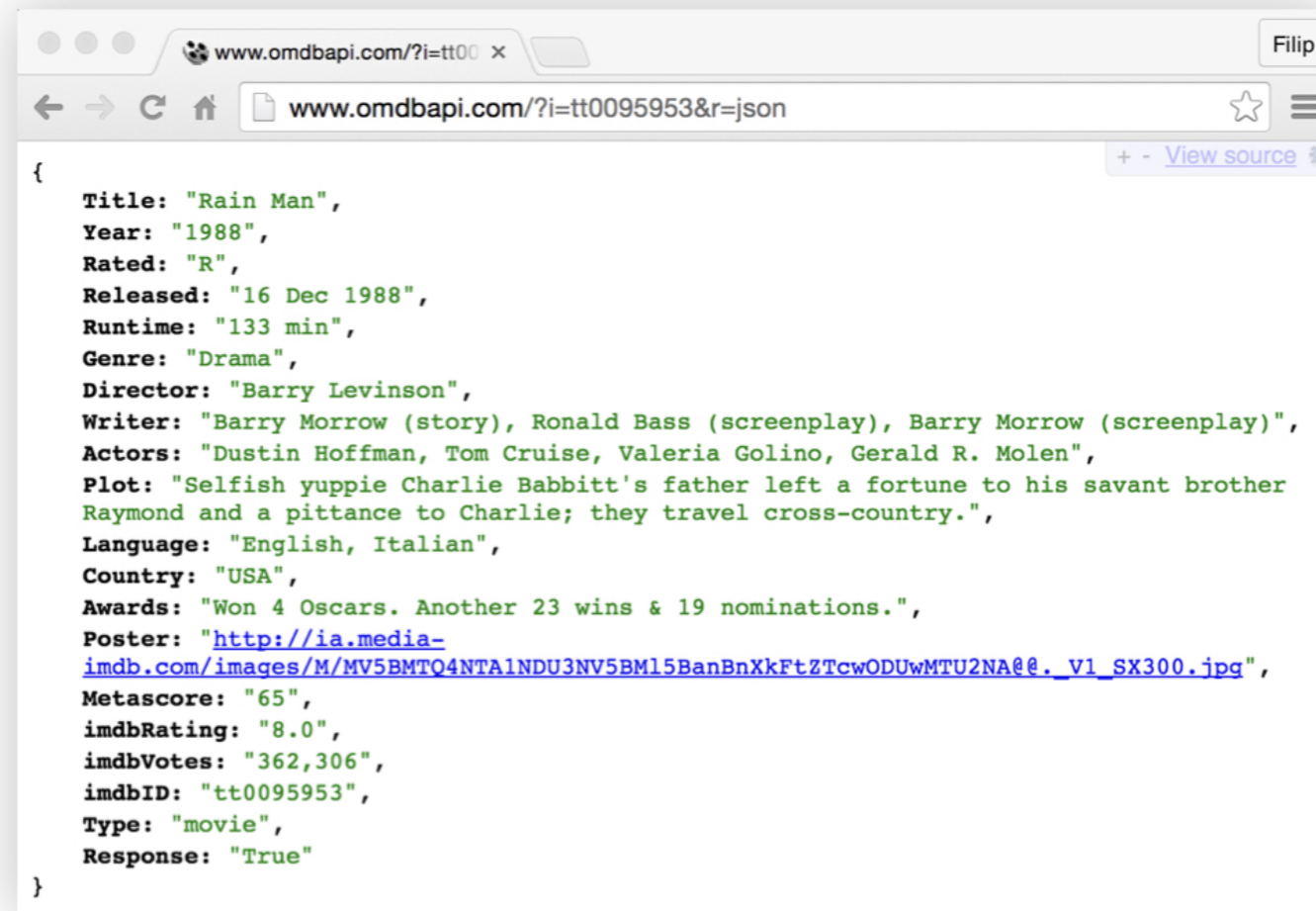
Info on Rain Man (1988)

```
url <- "http://www.imdb.com/title/tt0095953/"  
download.file(url, "local_imdb.html")
```

```
<div class="pro-title-link text-center">  
<a href="http://pro.imdb.com/title/tt0095953?rf=cons_tt_contact&ref_=cons_tt_conta">Contact the Filmmakers on IMDbPro &raquo;</a>  
</div> </td>  
  <td id="overview-top">  
    <div id="prometer_container">  
      <div id="prometer" class="meter-collapsed up">  
        <div id="meterHeaderBox">  
          <div id="meterTitle" class="meterToggleOnHover">Popularity</div>  
          <span id="meterRank">1,303</span>  
        </div>  
        <div id="meterChangeRow" class="meterToggleOnHover">  
          <span>Up</span>  
          <span id="meterChange">163</span>  
          <span>this week</span>  
        </div>  
      </div>  
    </div>  
<h1 class="header"> <span class="itemprop" itemprop="name">Rain Man</span>  
  <span class="nobr">( <a href="/year/1988/?ref_=tt_ov_inf">  
>1988</a> ) </span>
```

Rain Man JSON (OMDb API)

<http://www.omdbapi.com/?i=tt0095953&r=json>



The screenshot shows a web browser window with the URL `www.omdbapi.com/?i=tt0095953&r=json`. The page content is a JSON object representing the movie "Rain Man". The JSON data is as follows:

```
{
  Title: "Rain Man",
  Year: "1988",
  Rated: "R",
  Released: "16 Dec 1988",
  Runtime: "133 min",
  Genre: "Drama",
  Director: "Barry Levinson",
  Writer: "Barry Morrow (story), Ronald Bass (screenplay), Barry Morrow (screenplay)",
  Actors: "Dustin Hoffman, Tom Cruise, Valeria Golino, Gerald R. Molen",
  Plot: "Selfish yuppie Charlie Babbitt's father left a fortune to his savant brother Raymond and a pittance to Charlie; they travel cross-country.",
  Language: "English, Italian",
  Country: "USA",
  Awards: "Won 4 Oscars. Another 23 wins & 19 nominations.",
  Poster: "http://ia.media-imdb.com/images/M/MV5BMTQ4NTA1NDU3NV5BM15BanBnXkFtZTcwODUwMTU2NA@@._V1_SX300.jpg",
  Metascore: "65",
  imdbRating: "8.0",
  imdbVotes: "362,306",
  imdbID: "tt0095953",
  Type: "movie",
  Response: "True"
}
```

jsonlite

- Jeroen Ooms
- Improvement of earlier packages
- Consistent, robust
- Support all use-cases

Rain Man list in R

```
install.packages("jsonlite")  
library(jsonlite)  
fromJSON("http://www.omdbapi.com/?i=tt0095953&r=json")
```

```
List of 20  
 $ Title      : chr "Rain Man"  
 $ Year       : chr "1988"  
 $ Rated      : chr "R"  
 $ Released   : chr "16 Dec 1988"  
 $ Runtime    : chr "133 min"  
 ...  
 $ imdbVotes  : chr "359,903"  
 $ imdbID     : chr "tt0095953"  
 $ Type       : chr "movie"  
 $ Response   : chr "True"
```

- Way more structure!

Let's practice!

INTERMEDIATE IMPORTING DATA IN R

JSON & jsonlite

INTERMEDIATE IMPORTING DATA IN R



Filip Schouwenaars
Instructor, DataCamp

JSON object

```
{"id":1, "name":"Frank", "age":23, "married":false}
```

name	value
string	string
	number
	boolean
	null
	JSON object
	JSON array

JSON object

```
{"id":1,"name":"Frank","age":23,"married":false}
```

```
x <- '{"id":1,"name":"Frank","age":23,"married":false}'  
r <- fromJSON(x)  
str(r)
```

```
List of 4  
 $ id      : int 1  
 $ name    : chr "Frank"  
 $ age     : int 23  
 $ married: logi FALSE
```

JSON array

JSON

```
[4, 7, 4, 6, 4, 5, 10, 6, 6, 8]
```

```
fromJSON('[4, 7, 4, 6, 4, 5, 10, 6, 6, 8]')
```

```
4 7 4 6 4 5 10 6 6 8
```

JSON

```
[4, "a", 4, 6, 4, "b", 10, 6, false, null]
```

```
fromJSON('[4, "a", 4, 6, 4, "b", 10, 6, false, null]')
```

```
"4" "a" "4" "6" "4" "b" "10" "6" "FALSE" NA
```

JSON Nesting

```
{  
  "id": 1,  
  "name": "Frank",  
  "age": 23,  
  "married": false,  
  "partner": {  
    "id": 4,  
    "name": "Julie"  
  }  
}
```

JSON Nesting

```
r <- fromJSON('{ "id":1, "name":"Frank", "age":23,
                "married":false, "partner":{"id":4, "name":"Julie"}}')
Rstr(r)
```

```
List of 5
 $ id      : int 1
 $ name    : chr "Frank"
 $ age     : int 23
 $ married: logi FALSE
 $ partner: List of 2
  ..$ id   : int 4
  ..$ name: chr "Julie"
```

JSON Array of JSON Objects

```
[  
  {"id":1, "name":"Frank"},  
  {"id":4, "name":"Julie"},  
  {"id":12, "name":"Zach"}  
]
```

```
RfromJSON(' [{"id":1, "name":"Frank"},  
              {"id":4, "name":"Julie"},  
              {"id":12, "name":"Zach"} ]')
```

```
id name  
1  1 Frank  
2  4 Julie  
3 12 Zach
```


Other jsonlite functions

- `toJSON()`
- `prettyfy()`
- `minify()`

Let's practice!

INTERMEDIATE IMPORTING DATA IN R