

# Introduction to Web Scraping with R

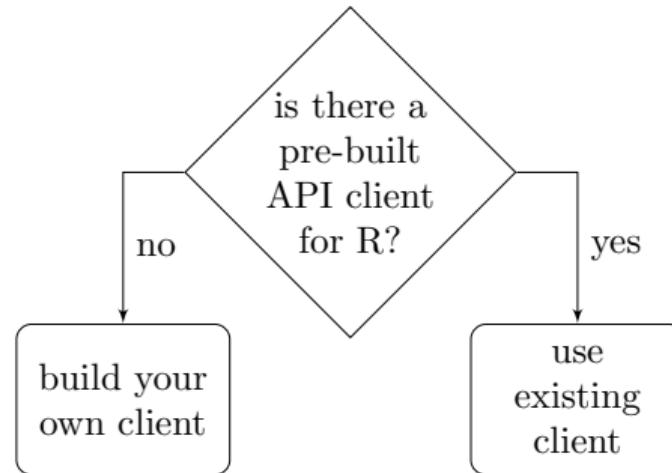
Accessing APIs from Scratch



Simon Munzert | IPSDS

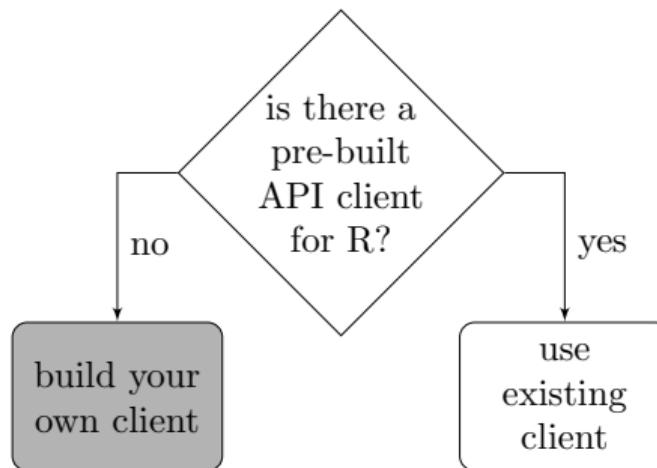
# API access with R

There are basically **two scenarios**:



# API access with R

There are basically **two scenarios**:



Here, we talk about the case where you have to build your own API binding.

# Accessing APIs from Scratch

# Accessing APIs from Scratch

Steps to be taken

# Accessing APIs from Scratch

## Steps to be taken

1. figure out how the API works: every API has a human-readable documentation for developers

# Accessing APIs from Scratch

## Steps to be taken

1. figure out how the API works: every API has a human-readable documentation for developers
2. build access to the API via R

# Accessing APIs from Scratch

## Steps to be taken

1. figure out how the API works: every API has a human-readable documentation for developers
2. build access to the API via R
3. build functionality that processes the data output (e.g., turns it into R objects)

# Example

# Example

## The IP API

- available at  
<http://ip-api.com/>
- its purpose: parses your IP  
(or a bunch of given IPs)  
and returns some values,  
including guessed location,  
service provider, and  
organization behind the IP

IP-API.com - Free Geolocation

ip-api.com

Simon

Query IP/domain

Your current IP Address

IP [REDACTED]

Country Germany

Country code DE

Region Land Berlin

Region code BE

City Berlin

Zip Code 12529

Latitude 52.5167

Longitude 13.4

Timezone Europe/Berlin

ISP Vodafone Kabel Deutschland

Organization Vodafone Kabel Deutschland

AS number/name AS31334 Vodafone Kabel Deutschland GmbH

TCP/IP fingerprint [REDACTED], Mac OS X

User-Agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

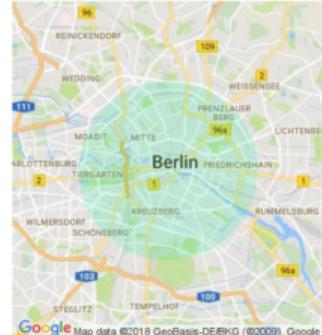
Internal IP [REDACTED]

DNS server [REDACTED] (Germany - Vodafone Kabel Deutschland)

API documentation

English - Română - Русский - Français - Deutsch - Español - Español (Argentina) - 中國 - 日本語

© 2017 IP-API.com | api | pro | contact



# Example

## The IP API

- available at  
<http://ip-api.com/>
- its purpose: parses your IP (or a bunch of given IPs) and returns some values, including guessed location service provider, and organization behind the IP
- check out API documentation

IP-API.com - Free Geolocation

ip-api.com

Simon

### IP-API.com Geolocation API

Query IP/domain

Your current IP Address

IP [REDACTED]

Country Germany

Country code DE

Region Land Berlin

Region code BE

City Berlin

Zip Code 12529

Latitude 52.5167

Longitude 13.4

Timezone Europe/Berlin

ISP Vodafone Kabel Deutschland

Organization Vodafone Kabel Deutschland

AS number/name AS31334 Vodafone Kabel Deutschland GmbH

TCP/IP fingerprint [REDACTED], Mac OS X

User-Agent Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_13\_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/63.0.3239.108 Safari/537.36

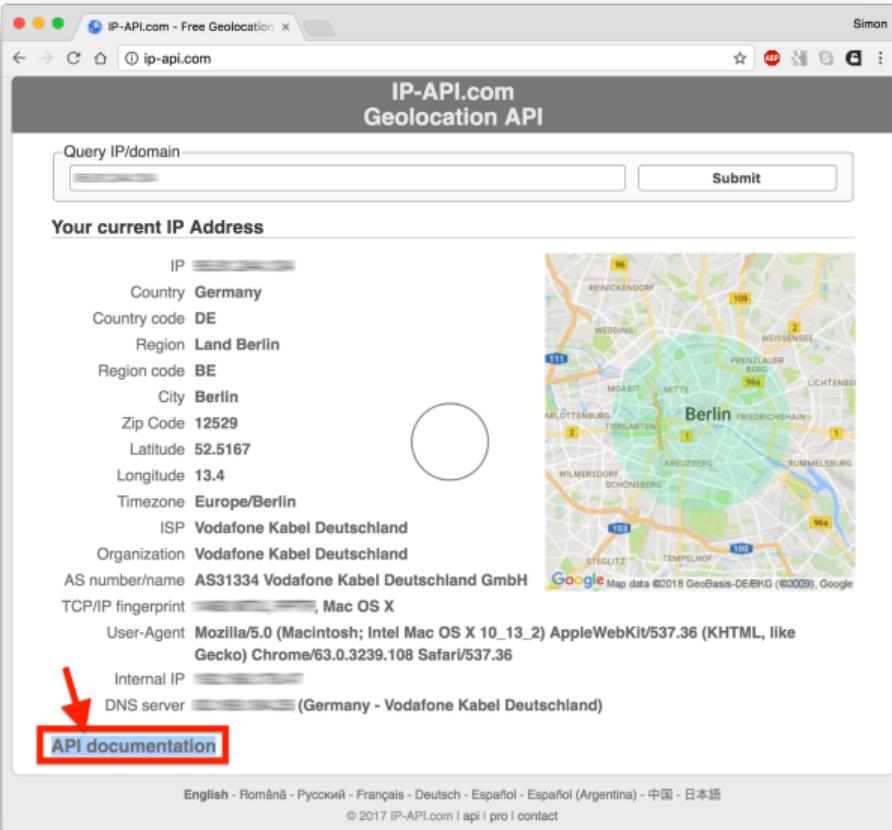
Internal IP [REDACTED]

DNS server [REDACTED] (Germany - Vodafone Kabel Deutschland)

[API documentation](#)

English - Română - Русский - Français - Deutsch - Español - Español (Argentina) - 中國 - 日本語

© 2017 IP-API.com | api | pro | contact



Berlin

Map data ©2018 GeoBasis-DE/BKG (2009), Google

# Example

## The IP API

- access is free for non-commercial use (up to 150 requests per minute)
- various response formats available
- paid pro service available

The screenshot shows a web browser window with the title "IP Geolocation API [ip-api]" and the URL "ip-api.com/docs/". The page content is as follows:

**IP Geolocation API**

**About**  
ip-api.com provides free usage of its Geo IP [API](#) through multiple response formats.

We support IPv4 and IPv6.

**Response formats and examples**

- [XML](#)
- [JSON](#)
- [CSV](#)
- [Newline Separated](#)
- [Serialized PHP](#)

**Usage limits**  
Our system will automatically ban any IP addresses doing over 150 requests per minute. To unban your IP click [here](#).

You are free to use ip-api.com for non-commercial use. **We do not allow commercial use without prior approval.**

For commercial, unlimited use see our [pro service](#).

Table of Contents

- IP Geolocation API
- About
- Response formats and examples
- Usage limits

Back to top

# Example

## The IP API

- to access the API, we have to send a [GET](#) request to <http://ip-api.com/json>
- we can provide any IP we want
- the response is raw JSON code

The screenshot shows a web browser displaying the documentation for the IP Geolocation API. The URL is <http://ip-api.com/docs/api:json>. The page title is "ip-api" and the sub-page title is "Page: • JSON". On the left, there's a sidebar with a navigation tree under "api" (CSV, Error messages, JSON, Batch JSON, Newline Separated, Return values, Serialized PHP, XML, Change Log, Statistics, DNS API, IP Geolocation API, Unban IP). To the right, there are sections for "JSON", "Usage", and "Response". The "Usage" section explains how to send a GET request to <http://ip-api.com/json> and mentions that you can supply an IP address or domain to lookup, or none to use your current IP address. The "Response" section shows a JSON object structure with fields like status, country, countryCode, region, regionName, city, zip, lat, lon, timezone, isp, org, as, and query. Below the JSON structure, it says "List of returned values". At the bottom, it says "A failed request will return, by default, the following:" followed by a placeholder JSON object. A "Table of Contents" sidebar on the right lists various API-related topics.

```
{
  "status": "success",
  "country": "COUNTRY",
  "countryCode": "COUNTRY CODE",
  "region": "REGION CODE",
  "regionName": "REGION NAME",
  "city": "CITY",
  "zip": "ZIP CODE",
  "lat": LATITUDE,
  "lon": LONGITUDE,
  "timezone": "TIME ZONE",
  "isp": "ISP NAME",
  "org": "ORGANIZATION NAME",
  "as": "AS NUMBER / NAME",
  "query": "IP ADDRESS USED FOR QUERY"
}
```

List of returned values

A failed request will return, by default, the following:

# Example

## IP API access in R

A **GET** request essentially means that we assemble a URL using

- the API **endpoint** (a basic URL) and
- **parameters-value pairs** that are added to the URL
- here, we only add the literal IP address
- we use **fromJSON** to pose the request and directly parse the data from the API

R code

---

```
1 url <- "http://ip-api.com/json"
2 url_ip <- paste0(url, "/208.80.152.201")
3 ip_parsed <- jsonlite::fromJSON(url_ip)
```

---

end

# Example

## Investigating the output

R code —

```
4 names(ip_parsed)
[1] "as"           "city"          "country"        "countryCode"   "isp"
[6] "lat"          "lon"           "org"            "query"         "region"
[11] "regionName"  "status"        "timezone"       "zip"

5 ip_parsed
$as
[1] "AS14907 Wikimedia Foundation, Inc."

$city
[1] "Cleveland"

$country
[1] "United States"

$countryCode
[1] "US"

$isp
```

# Example

## Bringing it into shape

- in our case, `fromJSON()` has created an R list object
- we turn it into a data frame

R code —

```
6 ip_parsed %>% as.data.frame(stringsAsFactors = FALSE)
              as      city      country countryCode
1 AS14907 Wikimedia Foundation, Inc. Cleveland United States          US
                  isp      lat      lon               org
1 Wikimedia Foundation, Inc. 41.4995 -81.6954 Wikimedia Foundation, Inc.
            query region regionName  status      timezone    zip
1 208.80.152.201      OH        Ohio success America/New_York 44192

```

— end

# Summary

# Summary

- accessing APIs from scratch can be almost as easy as using a readily available R client
- often however, APIs are more complex (provide much more parameters and variations in data output)
- ideally, you build functions around your API calls to make it even more accessible

