The `index.json` file is just a string of metadata. It looks like this:

```
{"basePath":"/covid19-mobility-data/2008HotfixDev38/v3",
 "mobilityDataVersion":"2008HotfixDev38:2020-05-21",
 "regions":{"en-us":{"jsonPath":"/en-us/applemobilitytrends.json",
                     "localeNamesPath":"/en-us/locale-names.json",
                     "csvPath":"/en-us/applemobilitytrends-2020-05-21.csv",
                     "initialPath":"/en-us/initial-data.json",
                     "shards":{"defaults":"/en-us/shards/defaults.json"}}}}
```

So, we grab this file (whose URL we know) and extract the information we want about the `basePath` and `csvPath` that point to the data:

```
get_apple_target <- function(cdn_url = "https://covid19-static.cdn-apple.com",
                             json_file = "covid19-mobility-data/current/v3
/index.json") {
  tf <- tempfile(fileext = ".json")
  curl::curl_download(paste0(cdn_url, "/", json_file), tf)
  json_data <- jsonlite::fromJSON(tf)
  paste0(cdn_url, target$basePath, target$regions$`en-us`$csvPath)
}


## > get_apple_target()
## [1] "https://covid19-static.cdn-apple.com/covid19-mobility-data/2008HotfixDev38/v3/en-us/
applemobilitytrends-2020-05-21.csv"
```

Then we can grab the data itself, with this function:

```


get_apple_data <- function(url = get_apple_target(),
                           fname = "applemobilitytrends-",
                           date = stringr::str_extract(get_apple_target(),
"\\d{4}-\\d{2}-\\d{2}"),
                           ext = "csv",
                           dest = "data-raw/data",
                           save_file = c("n", "y")) {

  save_file <- match.arg(save_file)
  message("target: ", url)

  destination <- fs::path(here::here("data-raw/data"),
                          paste0("apple_mobility", "_daily_", date), ext = ext)

  tf <- tempfile(fileext = ext)
  curl::curl_download(url, tf)
```

```
  ## We don't save the file by default
  switch(save_file,
         y = fs::file_copy(tf, destination),
         n = NULL)

  janitor::clean_names(readr::read_csv(tf))
}
```

This will pull the data into a tibble, which you can then clean further (e.g., put into long format) as desired.

```


apple <- get_apple_data()

## target: https://covid19-static.cdn-apple.com/covid19-mobility-data/2008HotfixDev38/v3/en-us/
applemobilitytrends-2020-05-21.csv
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   geo_type = col_character(),
##   region = col_character(),
##   transportation_type = col_character(),
##   alternative_name = col_character(),
##   `sub-region` = col_character(),
##   country = col_character(),
##   `2020-05-11` = col_logical(),
##   `2020-05-12` = col_logical()
## )
## See spec(...) for full column specifications.

apple

### A tibble: 3,625 x 136
##   geo_type region transportation_… alternative_name sub_region country
x2020_01_13 x2020_01_14 x2020_01_15
##
## 1 country… Alban… driving          NA               NA         NA
100        95.3       101.
## 2 country… Alban… walking          NA               NA         NA
100       101.         98.9
## 3 country… Argen… driving          NA               NA         NA
100        97.1       102.
## 4 country… Argen… walking          NA               NA         NA
100        95.1       101.
## 5 country… Austr… driving          AU               NA         NA
100       103.        104.
## 6 country… Austr… transit          AU               NA         NA
100       102.        101.
## 7 country… Austr… walking          AU               NA         NA
100       101.        102.
## 8 country… Austr… driving          Österreich       NA         NA
100       101.        104.
## 9 country… Austr… walking          Österreich       NA         NA
100       102.        106.
```

```
##10 country… Belgi… driving          België|Belgique  NA         NA
100       101.        107.
### … with 3,615 more rows, and 127 more variables: x2020_01_16 , x2020_01_17 ,
x2020_01_18 ,
###   x2020_01_19 , x2020_01_20 , x2020_01_21 , x2020_01_22 , x2020_01_23 ,
###   x2020_01_24 , x2020_01_25 , x2020_01_26 , x2020_01_27 , x2020_01_28 ,
###   x2020_01_29 , x2020_01_30 , x2020_01_31 , x2020_02_01 , x2020_02_02 ,
###   x2020_02_03 , x2020_02_04 , x2020_02_05 , x2020_02_06 , x2020_02_07 ,
###   x2020_02_08 , x2020_02_09 , x2020_02_10 , x2020_02_11 , x2020_02_12 ,
###   x2020_02_13 , x2020_02_14 , x2020_02_15 , x2020_02_16 , x2020_02_17 ,
###   x2020_02_18 , x2020_02_19 , x2020_02_20 , x2020_02_21 , x2020_02_22 ,
###   x2020_02_23 , x2020_02_24 , x2020_02_25 , x2020_02_26 , x2020_02_27 ,
###   x2020_02_28 , x2020_02_29 , x2020_03_01 , x2020_03_02 , x2020_03_03 ,
###   x2020_03_04 , x2020_03_05 , x2020_03_06 , x2020_03_07 , x2020_03_08 ,
###   x2020_03_09 , x2020_03_10 , x2020_03_11 , x2020_03_12 , x2020_03_13 ,
###   x2020_03_14 , x2020_03_15 , x2020_03_16 , x2020_03_17 , x2020_03_18 ,
###   x2020_03_19 , x2020_03_20 , x2020_03_21 , x2020_03_22 , x2020_03_23 ,
###   x2020_03_24 , x2020_03_25 , x2020_03_26 , x2020_03_27 , x2020_03_28 ,
###   x2020_03_29 , x2020_03_30 , x2020_03_31 , x2020_04_01 , x2020_04_02 ,
###   x2020_04_03 , x2020_04_04 , x2020_04_05 , x2020_04_06 , x2020_04_07 ,
###   x2020_04_08 , x2020_04_09 , x2020_04_10 , x2020_04_11 , x2020_04_12 ,
###   x2020_04_13 , x2020_04_14 , x2020_04_15 , x2020_04_16 , x2020_04_17 ,
###   x2020_04_18 , x2020_04_19 , x2020_04_20 , x2020_04_21 , x2020_04_22 ,
###   x2020_04_23 , x2020_04_24 , …
##
```