

# Package ‘openFDA’

May 9, 2026

**Title** 'openFDA' API

**Version** 0.1.0

**Description** The 'openFDA' API facilitates access to Federal Drug Agency (FDA) data on drugs, devices, foodstuffs, tobacco, and more with 'httr2'. This package makes the API easily accessible, returning objects which the user can convert to JSON data and parse. Kass-Hout TA, Xu Z, Mohebbi M et al. (2016) <[doi:10.1093/jamia/ocv153](https://doi.org/10.1093/jamia/ocv153)>.

**License** GPL (>= 3)

**URL** <https://github.com/simpar1471/openFDA>,  
<https://simpar1471.github.io/openFDA/>

**BugReports** <https://github.com/simpar1471/openFDA/issues>

**Imports** cli, httr2, checkmate, purrr, rlang, vctrs

**Config/Needs/website** rmarkdown

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), stringr

**Config/testthat/edition** 3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Simon Parker [aut, cre, cph] (ORCID:  
<<https://orcid.org/0009-0003-8214-4496>>)

**Maintainer** Simon Parker <[simon.parker.24@ucl.ac.uk](mailto:simon.parker.24@ucl.ac.uk)>

**Repository** CRAN

**Date/Publication** 2024-10-18 10:50:30 UTC

## Contents

format_search_term . . . . .	2
format_sort_term . . . . .	3
openFDA . . . . .	4
set_api_key . . . . .	7

---

format_search_term	<i>Format character vectors into search terms for openFDA API queries</i>
--------------------	---

---

### Description

This function is a helper for constructing search queries. Whilst it handles some of the available formatting for openFDA APIs, it does not recapture all of the search term syntax available to you. To get a full appreciation of the openFDA search syntax, see <https://open.fda.gov/apis/advanced-syntax/>.

### Usage

```
format_search_term(search, exact = TRUE, mode = "or")
```

### Arguments

search	A character vector of length 1 or more. If scalar and unnamed, it will be assumed that you have already formatted your search string to work with the API. If named, the vector will be collapsed to include your various search terms, separated by AND or OR terms based on the value of operator.
exact	A single-length logical vector. When TRUE (the default), individual search terms will be surrounded with double quotes (""). Set exact to FALSE if your search term contains multiple words to be searched on, e.g. <code>c("openfda.generic_name" = "losartan+candesartan")</code> . This parameter only applies if search is a named character vector.
mode	A single-length character vector, which defines how searches in multiple fields should be combined. By default ("or") they will be combined with an 'OR' operator, but you can make an 'AND' operator be used instead ("and"). This argument is case-sensitive and will throw an error if mode is not one of either "and" or "or". This parameter only applies if search is a named character vector.

### Value

A character vector of the S3 class `<AsIS>`, with a formatted search term which can be supplied to `openFDA()`.

### Note

This function does not check that you're providing accurate field names or search terms. It is up to you to make sure you've provided correctly spelt fields and search terms.

**See Also**

- `format_sort_term()` performs similar formatting for the sort component of an openFDA query.
- `I()` generates vectors with the <AsIs> S3 class.
- `httr2::req_url()` documents why `I()` is applied to the output of this function.

**Examples**

```
# Provide a formatted search string and the function will do no formatting
format_search_term("openfda.generic_name:verapamil")

# Provide a named vector and the function will format it for you
format_search_term(c("openfda.generic_name" = "verapamil"))

# If providing multiple elements in your search term, use `exact = FALSE`
# to prevent the function from surrounding the term with double quotes.
format_search_term(c("openfda.generic_name" = "verapamil+amlodipine"),
                  exact = FALSE)

# Provide a longer named vector and function will merge these with an OR
# operator
format_search_term(c("openfda.generic_name" = "verapamil",
                  "openfda.manufacturer_name" = "glaxo*"))

# Or you can set the `mode` argument to merge your search terms with an AND
# operator
format_search_term(c("openfda.generic_name" = "verapamil",
                  "openfda.manufacturer_name" = "glaxo*"),
                  mode = "and")
```

---

format\_sort\_term

*Format character vectors into sort terms for openFDA API queries*


---

**Description**

This function acts as a helper for constructing a sort term in the openFDA API.

**Usage**

```
format_sort_term(sort)
```

**Arguments**

`sort` A single-length character vector of length 1. If unnamed, it will be assumed that you have already formatted your search string to work with the API. If named, the vector will be collapsed to include your field and sorting choice.

**Value**

A character vector of the S3 class `<AsIS>`, with a formatted search term which can be supplied to `openFDA()`.

**Note**

This function does not check that you're providing accurate field names or search terms. It is up to you to make sure you've provided correctly spelt fields and search terms.

**See Also**

- `format_search_term()` performs similar formatting for the search component of an openFDA query.
- `I()` generates vectors with the `<AsIS>` S3 class.
- `httr2::req_url()` documents why `I()` is applied to the output of this function.

**Examples**

```
# Provide a formatted search string and the function will do no formatting
format_sort_term("openfda.generic_name:asc")

# Provide a named vector and the function will format it for you
format_sort_term(c("openfda.generic_name" = "asc"))

# Errors will be thrown if you supply a bad input
try(format_sort_term("receivedate:no_order"))
try(format_sort_term(c("receivedate" = "ascending")))
```

---

openFDA

*Send requests to the openFDA API*

---

**Description**

Send requests to the openFDA API

**Usage**

```
openFDA(
  search = "",
  sort = NULL,
  count = NULL,
  limit = 1000,
  skip = NULL,
  endpoint = "drug-drugsfda",
  api_key = get_api_key(),
  warn_on_http_error = TRUE
)
```

**Arguments**

search	A character vector which will be passed to <code>format_search_term()</code> . If search is "" (the default), openFDA will retrieve all records with no filtering. An error will be thrown if any elements of search are missing (NA).
sort	A single string or scalar named character vector describing how to sort the results. The sort term should either be formatted as "[FIELD]:[asc/desc]", or c("[FIELD]" = "[asc/desc]"). For example, to sort results in the Drugs@FDA endpoint by ascending submission status dates. If values other than "asc", or "desc" are supplied, the function will throw an error.
count	A single string denoting a field on which to count results. If NULL (the default), results will be returned in full to the user. Specify this parameter if you want to count occurrences of results within your search term - go to <a href="https://open.fda.gov/apis/query-syntax/">https://open.fda.gov/apis/query-syntax/</a> for more information.
limit	A single integerish value describing the limit on the number of records to retrieve. An error will be thrown if limit is more than 1000 (the default).
skip	A single integer describing how many records should be skipped. If more records are skipped than are found in your search, the openFDA API will return a 404 error.
endpoint	A single-length character vector describing which openFDA endpoint to target. <ul style="list-style-type: none"> <li>• "animal-event": Adverse event reports for animal/veterinary drugs.</li> <li>• "drug-event": Adverse event reports from <a href="#">FDA Adverse Event Reporting System</a></li> <li>• "drug-label": Drug documentation in the <a href="#">Structured Product Labeling (SPL)</a> format.</li> <li>• "drug-ndc": Data from the <a href="#">National Drug Code Directory (NDC)</a>.</li> <li>• "drug-enforcement": Data from the <a href="#">FDA Recall Enterprise System</a> about drug products.</li> <li>• "drug-drugsfda": Data on products approved for human use by the FDA since 1939, with mostly complete data after 1998.</li> <li>• "device-510k": Data from 510(k) submissions to the FDA regarding medical devices.</li> <li>• "device-classification": Data from the FDA <a href="#">Product Classification Database</a> for medical devices.</li> <li>• "device-enforcement": Data from the <a href="#">FDA Recall Enterprise System (RES)</a> for medical devices.</li> <li>• "device-event": Data on adverse events associated with FDA-approved medical devices.</li> <li>• "device-pma": Pre-market approval data for medical devices submitted to the FDA.</li> <li>• "device-recall": Data on product recalls for medical devices which violate FDA law.</li> <li>• "device-reglist": Data on FDA <a href="#">Device Registrations and Listings</a>.</li> <li>• "device-covid19serology": Data from <a href="#">independent evaluations of COVID-19 serological tests</a>.</li> </ul>

- "device-udi": Data from the FDA's Global Unique Device Identification Database (GUDID).
- "food-enforcement": Data from the [FDA Recall Enterprise System](#) for food products.
- "food-event": Data from the [Center for Food Safety and Applied Nutrition Adverse Event Reporting System](#).
- "other-historicaldocument": The majority of FDA Press Releases, from 1913 to 2014.
- "other-nsde": Data from the [National Drug Code Structured Product Labeling Data Elements](#).
- "other-substance": Data regarding substances - individual molecules, proteins, nucleic acids, and more.
- "other-unii": Data from the [FDA's Global Substance Registration System](#).
- "tobacco-problem": Data on problems (e.g. damage, defects, contamination, bad smell) with tobacco products.

This argument is case-sensitive. By default, the package will target the `Drugs@FDA` endpoint (`"drugs-drugsfda"`).

<code>api_key</code>	A single-length character vector with your openFDA API key. By default this is the result of <code>get_api_key()</code> . If <code>api_key</code> is an empty string, an error will be thrown.
<code>warn_on_http_error</code>	A scalar logical value. If TRUE (the default), common openFDA HTTP errors will cause explanatory warnings to be printed. If FALSE, the underlying <code>httr2</code> response object will be returned with no extra warnings.

### Value

An `httr2` response object from `httr2::req_perform()`. You can use `httr2::resp_body_json()` to extract JSON data from the response.

### References

Kass-Hout TA, Xu Z, Mohebbi M, Nelsen H, Baker A, Levine J, Johansen E, Bright RA. **OpenFDA: an innovative platform providing access to a wealth of FDA's publicly available data** *J Am Med Inform Assoc* 2016, **23(3):596-600**. doi:[10.1093/jamia/ocv153](https://doi.org/10.1093/jamia/ocv153)

### See Also

[format\\_search\\_term\(\)](#) documents how input search vectors are converted to openFDA API searches.

### Examples

```
if (httr2::secret_has_key("OPENFDA_KEY")) {
  set_api_key(httr2::secret_decrypt(
    "TEaDtqdFMq9_Montij5p9IY6T57IyqkbF8IYFV0pk-ttxotFUNdJSxgccAnkq4nQhplaf-r3deQ",
    "OPENFDA_KEY")
  )
}
```

```
))

resp <- openFDA(search = "openfda.manufacturer_name:gilead*",
                limit = 2,
                skip = 10)

# The function returns an `httr2` object
print(resp)
}

# Bad inputs will cause informative errors - here, a bad API key is supplied
try(
  openFDA(search = "openfda.manufacturer_name:gilead*",
           api_key = "BAD_API_KEY",
           limit = 1)
)
```

---

set\_api\_key

*Get and set your openFDA API keys*

---

## Description

Get and set your openFDA API keys

## Usage

```
set_api_key(api_key)
```

```
get_api_key()
```

## Arguments

`api_key` A single-length character vector with your openFDA API key. You can generate an API key on the [FDA website](#).

## Value

A single length character vector with your API key. For `set_api_key()`, this is returned invisibly. For `get_api_key()`, an error will be thrown if no key has been set.

## Note

To permanently set the API key for a given project, set `OPENFDA_TOKEN` in `.Renvirom`.

**Examples**

```
# Set your openFDA API key with `set_api_key()`  
api_key <- "example_api_key"  
set_api_key(api_key)  
  
# Retrieve it with `get_api_key()`  
get_api_key()  
  
# An error will be thrown if your API key is an empty string.  
set_api_key("")  
  
try(get_api_key())
```

# Index

`format_search_term`, 2  
`format_search_term()`, 4, 6  
`format_sort_term`, 3  
`format_sort_term()`, 3  
  
`get_api_key (set_api_key)`, 7  
  
`httr2::req_perform()`, 6  
`httr2::req_url()`, 3, 4  
`httr2::resp_body_json()`, 6  
  
`I()`, 3, 4  
  
`openFDA`, 4  
  
`set_api_key`, 7