Package: npi (via r-universe)

October 24, 2026

Title Access the U.S. National Provider Identifier Registry API

Version 0.3.1

Maintainer Frank Farach <frank.farach@gmail.com>

Description Access the United States National Provider Identifier Registry API <https://npiregistry.cms.hhs.gov/api/>. Obtain and transform administrative data linked to a specific individual or organizational healthcare provider, or perform advanced searches based on provider name, location, type of service, credentials, and other attributes exposed by the API.

License MIT + file LICENSE

URL https://github.com/ropensci/npi/, https://docs.ropensci.org/npi/, https://npiregistry.cms.hhs.gov/api/

BugReports https://github.com/ropensci/npi/issues/

Depends R (>= 3.1)

Imports checkLuhn, checkmate, curl, dplyr, glue, httr, magrittr, purrr, rlang, stringr, tibble, tidyr, utils

Suggests covr, httptest, knitr, mockery, rmarkdown, spelling, testthat (>= 2.1.0)

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libicu-dev libssl-dev

Repository https://ropensci.r-universe.dev

Date/Publication 2026-09-24 19:39:15 UTC

RemoteUrl https://github.com/ropensci/npi

RemoteRef master

RemoteSha f5eae3f74be8f4fa2a0fdd5774443a5f1f6c23e6

Contents

npi_flatten 2
npi_flatten.npi_results 3
npi_is_valid 4
npi_search 4
npi_summarize 7
npi_summarize.npi_results 8
npis 9

Index **11**

`npi_flatten` *S3 method to flatten an* `npi_results` *object*

Description

S3 method to flatten an `npi_results` object

Usage

```
npi_flatten(df, cols, key)
```

Arguments

`df`	A data frame containing the results of a call to `npi_search`.
`cols`	If non-NULL, only the named columns specified here will be be flattened and returned along with `npi`.
`key`	A quoted column name from `df` to use as a matching key. The default value is `"npi"`.

Details

The names of unnested columns are prefixed by the name of their originating list column to avoid name clashes and show their lineage. List columns containing all NULL data will be absent from the result because there are no columns to unnest. All input keys are retained even when their selected nested data is empty, regardless of the order of `cols`. If all selected nested data is empty, only the keys are returned. When multiple nested tables have more than one row for a key, their rows are combined in all possible combinations.

Value

A data frame (tibble) with flattened list columns.

Examples

```
# Flatten all list columns
data(npis)
npi_flatten(npis)

# Only flatten specified columns
npi_flatten(npis, cols = c("basic", "identifiers"))
```

`npi_flatten.npi_results` *Flatten NPI search results*

Description

This function takes an `npi_results` S3 object returned by `npi_search` and flattens its list columns. It unnests the lists columns and left joins them by `npi`. You can optionally specify which columns from `df` to include.

Usage

```
## S3 method for class 'npi_results'
npi_flatten(df, cols = NULL, key = "npi")
```

Arguments

`df`	A data frame containing the results of a call to `npi_search`.
`cols`	If non-NULL, only the named columns specified here will be be flattened and returned along with `npi`.
`key`	A quoted column name from `df` to use as a matching key. The default value is `"npi"`.

Details

The names of unnested columns are prefixed by the name of their originating list column to avoid name clashes and show their lineage. List columns containing all NULL data will be absent from the result because there are no columns to unnest. All input keys are retained even when their selected nested data is empty, regardless of the order of `cols`. If all selected nested data is empty, only the keys are returned. When multiple nested tables have more than one row for a key, their rows are combined in all possible combinations.

Value

A data frame (tibble) with flattened list columns.

Examples

```
# Flatten all list columns
data(npis)
npi_flatten(npis)

# Only flatten specified columns
npi_flatten(npis, cols = c("basic", "identifiers"))
```

`npi_is_valid` *Check if candidate NPI number is valid*

Description

Check whether a number is a valid NPI number per the specifications detailed in the Final Rule for the Standard Unique Health Identifier for Health Care Providers (69 FR 3434).

Usage

```
npi_is_valid(x)
```

Arguments

`x`	10-digit candidate NPI number

Value

Boolean indicating whether `npi` is valid

Examples

```
npi_is_valid(1234567893) # TRUE
npi_is_valid(1234567898) # FALSE
```

`npi_search` *Search the NPI Registry*

Description

Search the U.S. National Provider Identifier (NPI) Registry using parameters exposed by the registry's API (Version 2.1). Results are combined and returned as a tibble with an S3 class of `npi_results`. See `Value` below for a description of the returned object. API documentation may differ from what is shown here. Consult https://npiregistry.cms.hhs.gov/api-page for the latest documentation.

Usage

```
npi_search(
  number = NULL,
  enumeration_type = NULL,
  taxonomy_description = NULL,
  first_name = NULL,
  last_name = NULL,
  use_first_name_alias = NULL,
  organization_name = NULL,
  address_purpose = NULL,
  city = NULL,
  state = NULL,
  postal_code = NULL,
  country_code = NULL,
  limit = 10L
)
```

Arguments

number	(Optional) 10-digit NPI number assigned to the provider.
enumeration_type	(Optional) Type of provider associated with the NPI, one of: **"ind"** Individual provider (NPI-1) **"org"** Organizational provider (NPI-2)
taxonomy_description	(Optional) Scalar character vector with an exact description or exact specialty or wildcard * after 2 characters from the NUCC Healthcare Provider Taxonomy.
first_name	(Optional) This field only applies to Individual Providers. Trailing wildcard entries are permitted requiring at least two characters to be entered (e.g. "jo*"). This field allows the following special characters: ampersand, apostrophe, colon, comma, forward slash, hyphen, left and right parentheses, period, pound sign, quotation mark, and semi-colon.
last_name	(Optional) This field only applies to Individual Providers. Trailing wildcard entries are permitted requiring at least two characters to be entered. This field allows the following special characters: ampersand, apostrophe, colon, comma, forward slash, hyphen, left and right parentheses, period, pound sign, quotation mark, and semi-colon.
use_first_name_alias	(Optional) This field only applies to Individual Providers when not doing a wildcard search. When set to "True", the search results will include Providers with similar First Names. E.g., first_name=Robert, will also return Providers with the first name of Rob, Bob, Robbie, Bobby, etc. Valid Values are: TRUE: Will include alias/similar names; FALSE: Will only look for exact matches.
organization_name	(Optional) This field only applies to Organizational Providers. Trailing wildcard entries are permitted requiring at least two characters to be entered. This field allows the following special characters: ampersand, apostrophe, "at" sign, colon,

comma, forward slash, hyphen, left and right parentheses, period, pound sign, quotation mark, and semi-colon. Both the Organization Name and Other Organization Name fields associated with an NPI are examined for matching contents, therefore, the results might contain an organization name different from the one entered in the Organization Name criterion.

`address_purpose`
: Refers to whether the address information entered pertains to the provider's Mailing Address or the provider's Practice Location Address. When not specified, the results will contain the providers where either the Mailing Address or any of Practice Location Addresses match the entered address information. Primary will only search against Primary Location Address. While Secondary will only search against Secondary Location Addresses. Valid values are: "location", "mailing", "primary", "secondary".

`city`
: The City associated with the provider's address identified in Address Purpose. To search for a Military Address enter either APO or FPO into the City field. This field allows the following special characters: ampersand, apostrophe, colon, comma, forward slash, hyphen, left and right parentheses, period, pound sign, quotation mark, and semi-colon.

`state`
: The State abbreviation associated with the provider's address identified in Address Purpose. This field cannot be used as the only input criterion. If this field is used, at least one other field, besides the Enumeration Type and Country, must be populated. Valid values for states: https://npiregistry.cms.hhs.gov/registry/API-State-Abbr

`postal_code`
: The Postal Code associated with the provider's address identified in Address Purpose. If you enter a 5 digit postal code, it will match any appropriate 9 digit (zip+4) codes in the data. Trailing wildcard entries are permitted requiring at least two characters to be entered (e.g., "21*").

`country_code`
: The Country associated with the provider's address identified in Address Purpose. This field can be used as the only input criterion as long as the value selected is not US (United States). Valid values for country codes: https://npiregistry.cms.hhs.gov/registry/API-Country-Abbr

`limit`
: Maximum number of records to return, from 1 to 1200 inclusive. The default is 10. Because the API returns up to 200 records per request, values of `limit` greater than 200 will result in multiple API calls.

Details

By default, the function requests up to 10 records, but the `limit` argument accepts values from 1 to the API's limit of 1200.

Value

Data frame (tibble) containing the results of the search.

References

https://npiregistry.cms.hhs.gov/registry/help-api Data dictionary for fields returned

NUCC Healthcare Provider Taxonomy

Examples

```
## Not run:
# 10 NPI records for New York City
npi_search(city = "New York City")

# 10 NPI records for New York City, organizations only
npi_search(city = "New York City", enumeration_type = "org")

# 10 NPI records for New York City, individuals only
npi_search(city = "New York City", enumeration_type = "ind")

# 1200 NPI records for New York City
npi_search(city = "New York City", limit = 1200)

# Nutritionists in Maine
npi_search(state = "ME", taxonomy_description = "Nutritionist")

# Record associated with NPI 1245251222
npi_search(number = 1245251222)

## End(Not run)
```

---

npi_summarize *S3 method to summarize an* npi_results *object*

Description

S3 method to summarize an npi_results object

Usage

```
npi_summarize(object, ...)
```

Arguments

object	An npi_results S3 object
...	Additional optional arguments

Details

Missing optional second address lines are treated as empty strings. Missing required address components leave the full address missing.

Value

Tibble containing the following columns:

npi National Provider Identifier (NPI) number

`name` Provider's first and last name for individual providers, organization name for organizational providers.

`enumeration_type` Type of provider associated with the NPI, either "Individual" or "Organizational"

`primary_practice_address` Full address of the provider's primary practice location

`phone` Provider's telephone number

`primary_taxonomy` Primary taxonomy description. If no taxonomy is marked as primary for a record, the first listed taxonomy is used.

Examples

```
data(npis)
npi_summarize(npis)
```

---

npi_summarize.npi_results

Summary method for `npi_results` *S3 object*

Description

Print a human-readable overview of each record return in the results from a call to `npi_search`. The format of the summary is modeled after the one offered on the NPI registry website.

Usage

```
## S3 method for class 'npi_results'
npi_summarize(object, ...)
```

Arguments

`object`	An `npi_results` S3 object
`...`	Additional optional arguments

Details

Missing optional second address lines are treated as empty strings. Missing required address components leave the full address missing.

Value

Tibble containing the following columns:

`npi` National Provider Identifier (NPI) number

`name` Provider's first and last name for individual providers, organization name for organizational providers.

`enumeration_type` Type of provider associated with the NPI, either "Individual" or "Organizational"

`primary_practice_address` Full address of the provider's primary practice location

`phone` Provider's telephone number

`primary_taxonomy` Primary taxonomy description. If no taxonomy is marked as primary for a record, the first listed taxonomy is used.

Examples

```
data(npis)
npi_summarize(npis)
```

`npis` *Sample results from the NPI Registry*

Description

A dataset containing 10 records returned from an NPI Registry search for providers with a primary address in New York City.

Usage

```
npis
```

Format

A tibble with 10 rows and 11 columns, organized as follows:

npi [integer] 10-digit National Provider Identifier number

enumeration_type [character] Type of provider NPI, either "Individual" or "Organizational".

basic [list of 1 tibble] Basic information about the provider.

other_names [list of tibbles] Other names the provider goes by.

identifiers [list of tibbles] Other identifiers linked to the NPI.

taxonomies [list of tibbles] Healthcare Provider Taxonomy classification.

addresses [list of tibbles] Addresses for the provider's primary practice location and primary mailing address.

practice_locations [list of tibbles] Addresses for the provider's other practice locations.

endpoints [list of tibbles] Details about provider's endpoints for health information exchange.

created_date [datetime] Date NPI record was first created (UTC).

last_updated_date [datetime] UTC timestamp of the last time the NPI record was updated.

Details

```
search_npi(city = "New York City", limit = 10)
```

Source

https://npiregistry.cms.hhs.gov/registry/help-api

Index

∗ **data wrangling functions**
    npi_flatten, 2
∗ **datasets**
    npis, 9
∗ **search functions**
    npi_search, 4
∗ **summary functions**
    npi_summarize, 7
∗ **utility functions**
    npi_is_valid, 4

npi_flatten, 2
npi_flatten.npi_results, 3
npi_is_valid, 4
npi_search, *2*, *3*, 4, *8*
npi_summarize, 7
npi_summarize.npi_results, 8
npis, 9