# Package: eppoFindeR (via r-universe)

May 25, 2026

**Type** Package

**Title** Interface to the EPPO Database and Public APIs

**Version** 2.0.0

**Maintainer** Luca Belmonte <luca.belmonte@efsa.europa.eu>

**Description** Provides an interface to the public APIs of the European and Mediterranean Plant Protection Organization (EPPO) database. It enables users to retrieve EPPO data by accessing specific services and datasets. The package also includes utilities for data wrangling, including the integration of taxonomy with rank information. See <https://data.eppo.int/apis/> for more details.

**License** EUPL-1.2

**URL** https://openefsa.github.io/eppoFindeR/

**BugReports** https://github.com/openefsa/eppoFindeR/issues

**Depends** R (>= 4.1.0)

**Imports** cli (>= 3.6.5), httr2 (>= 1.2.1), jsonlite (>= 1.8.8), dplyr (>= 1.1.4), glue (>= 1.7.0), purrr (>= 1.0.2), checkmate (>= 2.3.1), tibble(>= 3.3.0)

**Suggests** devtools (>= 2.4.5), roxygen2 (>= 7.3.1), testthat (>= 3.2.1), usethis (>= 2.2.3), knitr (>= 1.0), rmarkdown (>= 2.0), covr (>= 3.6.4)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Config/Needs/website** pkgdown

**Roxygen** list(markdown = TRUE)

**Config/pak/sysreqs** libssl-dev

**Repository** https://openefsa.r-universe.dev

**Date/Publication** 2026-04-22 08:13:58 UTC

**RemoteUrl** https://github.com/openefsa/eppofinder

**RemoteRef** HEAD

**RemoteSha** 5143299434de92c387a03ab3fbf775653d09f0ae

# Contents

country . . . . . 2
general . . . . . 3
references . . . . . 4
reportingService . . . . . 5
rppo . . . . . 6
taxon . . . . . 8
taxons . . . . . 9
tools . . . . . 10
uniformTaxonomy . . . . . 11

**Index** **13**

---

`country` *Query the EPPO API Country endpoints.*

---

## Description

This function queries the Country endpoints of the EPPO Global Database via REST API for one or more ISO code(s) and one or more service(s). For each ISO code in `isoCodes`, the function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
country(
  isoCodes,
  services = c("overview", "categorization", "presence"),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

## Arguments

`isoCodes` `character` (vector). One or more ISO codes to query.

ISO country codes are standardized identifiers for countries and territories. These codes, typically consisting of two-letter, provide unique and consistent abbreviations of country names. The ISO coding system enables easy management, exchange, and integration of country information across databases and IT systems. ISO country codes and their related data can be accessed via the ISO Online Browsing Platform.

services — `character` (vector). One or more Country services to query. Common service names include: `overview`, `categorization` and `presence`. A validation step ensures that all provided services match the supported service names.

By default: all services.

apiKey — `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.

By default: Sys.getenv("EPPO_API_KEY").

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content for all the specified ISO codes.

## Examples

```
## Not run:
# Get all information about France.
countryData_ <- country(isoCodes = c("FR"))

# Get categorization data about France.
countryData_ <- country(isoCodes = c("FR"), services = c("categorization"))

# Get basic data about France and Italy.
countryData_ <- country(isoCodes = c("FR", "IT"), services = c("overview"))

## End(Not run)
```

---

`general` *Query the EPPO API General endpoint.*

---

## Description

This function queries the General endpoints of the EPPO Global Database via REST API. The function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
general(services = c("status"), apiKey = Sys.getenv("EPPO_API_KEY"))
```

## Arguments

services — `character` (vector). One or more General services to query. Common service names include: `status`. A validation step ensures that all provided services match the supported service names.

By default: all services.

apiKey — character (string). The API key used for authentication. It can be specified manually or through the .Renviron file.

By default: Sys.getenv("EPPO_API_KEY").

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content.

## Examples

```
## Not run:
# Get information about system health status.
generalData_ <- general()

## End(Not run)
```

---

# references — *Query the EPPO API References endpoints.*

---

## Description

This function queries the References endpoints of the EPPO Global Database via REST API. The function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
references(
  services = c("rppos", "qList", "distributionStatus", "pestHostClassification",
    "vectorClassification", "countries", "countriesStates"),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

## Arguments

services — character (vector). One or more References services to query. Common service names include: `rppos`, `qList`, `distributionStatus`, `pestHostClassification`, `vectorClassification`, `countries` and `countriesStates`. A validation step ensures that all provided services match the supported service names.

By default: all services.

apiKey — character (string). The API key used for authentication. It can be specified manually or through the .Renviron file.

By default: Sys.getenv("EPPO_API_KEY").

### Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content.

### Examples

```
## Not run:
# Get all information.
referencesData_ <- references()

# Get information about distribution status codes.
referencesData_ <- references(services = c("distributionStatus"))

# Get information about EPPO list codes and labels and countries.
referencesData_ <- references(services = c("qList", "countries"))

## End(Not run)
```

---

## reportingService — *Query the EPPO API Taxons endpoints.*

---

### Description

This function queries the Taxons endpoints of the EPPO Global Database via REST API. The function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

### Usage

```
reportingService(
  services = c("list", "reporting", "article"),
  params = list(list = list(), reporting = list(), article = list()),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

### Arguments

| | |
|---|---|
| `services` | `character` (vector). One or more Taxons services to query. Common service names include: `list`, `reporting` and `article`. A validation step ensures that all provided services match the supported service names.<br>By default: all services. |
| `params` | `list`. A named list of query parameters to include in the request. The list of available parameters can be accessed via the EPPO API Documentation platform.<br>By default: no parameters, you need to specify the required ones. |
| `apiKey` | `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.<br>By default: Sys.getenv("EPPO_API_KEY"). |

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content.

## Examples

```
## Not run:
# Get the list of reporting service issues.
reportingServiceData_ <- reportingService(
  services = c("list"),
  params = list(
    "list" = list()))

# Get a specific reporting service issue by ID.
reportingServiceData_ <- reportingService(
  services = c("reporting"),
  params = list(
    reporting = list(
      reporting_id = 10)))

# Get a specific article by ID.
reportingServiceData_ <- reportingService(
  services = c("article"),
  params = list(
    article = list(
      article_id = 234)))

# Get a specific reporting service issue and a specific article by ID.
reportingServiceData_ <- reportingService(
  services = c("reporting", "article"),
  params = list(
    reporting = list(reporting_id = 10),
    article = list(article_id = 234)
  ))

## End(Not run)
```

---

`rppo` *Query the EPPO API RPPO endpoints.*

---

## Description

This function queries the RPPO endpoints of the EPPO Global Database via REST API for one or more RPPO code(s) and one or more service(s). For each RPPO code in `rppoCodes`, the function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
rppo(
  rppoCodes,
  services = c("overview", "categorization"),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

## Arguments

| | |
|---|---|
| `rppoCodes` | `character` (vector). One or more RPPO codes to query. Available RPPO codes can be retrieved via the `references/rppos` service. |
| `services` | `character` (vector). One or more RPPO services to query. Common service names include: `overview` and `categorization`. A validation step ensures that all provided services match the supported service names.<br>By default: all services. |
| `apiKey` | `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.<br>By default: Sys.getenv("EPPO_API_KEY"). |

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content for all the specified RPPO codes.

## Examples

```
## Not run:
# Get all information about the European and Mediterranean Plant Protection
# Organisation.
rppoData_ <- rppo(rppoCodes = c("9A"))

# Get categorization data about the European and Mediterranean Plant
# Protection Organisation.
rppoData_ <- rppo(rppoCodes = c("9A"), services = c("categorization"))

# Get basic data about European and Mediterranean Plant Protection
# Organisation and the European Union.
rppoData_ <- rppo(rppoCodes = c("9A", "9L"), services = c("overview"))

## End(Not run)
```

taxon *Query the EPPO API Taxon endpoints.*

## Description

This function queries the Taxon endpoints of the EPPO Global Database via REST API for one or more EPPO code(s) and one or more service(s). For each EPPO code in `eppoCodes`, the function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
taxon(
  eppoCodes,
 services = c("overview", "infos", "names", "taxonomy", "categorization", "kingdom",
    "hosts", "pests", "vectors", "vectorof", "bca", "bcaof", "photos",
    "reporting_articles", "documents", "standards", "distribution"),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

## Arguments

`eppoCodes` `character` (vector). One or more EPPO codes to query.

EPPO codes are standardized identifiers for plants and pests (including pathogens). These alphanumeric codes, typically of 5 to 6 letters long, serve as unique, mnemonic abbreviations of scientific names. The EPPO coding system facilitates the management and exchange of organism names across databases and IT systems. EPPO codes and their corresponding data can be accessed via the EPPO Data Services platform.

`services` `character` (vector). One or more Taxon services to query. Common service names include: `overview`, `infos`, `names`, `taxonomy`, `categorization`, `kingdom`, `hosts`, `pests`, `vectors`, `vectorof`, `bca`, `bcaof`, `photos`, `reporting_articles`, `documents`, `standards` and `distribution`. A validation step ensures that all provided services match the supported service names.

By default: all services.

`apiKey` `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.

By default: Sys.getenv("EPPO_API_KEY").

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content for all the specified EPPO codes.

## Examples

```
## Not run:
# Get all information about Bemisia tabaci.
taxonData_ <- taxon(eppoCodes = c("BEMITA"))

# Get names data about Bemisia tabaci.
taxonData_ <- taxon(eppoCodes = c("BEMITA"), services = c("names"))

# Get taxonomy and categorization data about Bemisia tabaci and Gossypium
# hirsutum.
taxonData_ <- taxon(
  eppoCodes = c("BEMITA", "GOSHI"),
  services = c("taxonomy", "categorization"))

## End(Not run)
```

---

`taxons` *Query the EPPO API Taxons endpoints.*

---

## Description

This function queries the Taxons endpoints of the EPPO Global Database via REST API. The function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

## Usage

```
taxons(
  services = c("list"),
  params = list(list = list(limit = 100, offset = 0, orderAsc = TRUE, orderBy =
    "creation")),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

## Arguments

| | |
|---|---|
| `services` | `character` (vector). One or more Taxons services to query. Common service names include: `list`. A validation step ensures that all provided services match the supported service names.<br>By default: all services. |
| `params` | `list` (optional). A named list of query parameters to include in the request. The list of available parameters can be accessed via the EPPO API Documentation platform.<br>By default: default parameters. |
| `apiKey` | `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.<br>By default: Sys.getenv("EPPO_API_KEY"). |

### Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content.

### Examples

```
## Not run:
# Get the list of taxons.
taxonsData_ <- taxons()

# Get the list of taxons with custom parameters.
taxonsData_ <- taxons(params = list(
  "list" = list(
    createdFromDate = "2000-01-01",
    limit = 5,
    offset = 1,
    orderAsc = FALSE,
    orderBy = "eppocode")))

## End(Not run)
```

---

## tools — *Query the EPPO API Tools endpoints.*

---

### Description

This function queries the Tools endpoints of the EPPO Global Database via REST API. The function sequentially queries all specified `services` and returns the extracted data through a list of dataframes.

### Usage

```
tools(
  services = c("name2codes"),
  params = list(name2codes = list()),
  apiKey = Sys.getenv("EPPO_API_KEY")
)
```

### Arguments

| | |
|---|---|
| `services` | `character` (vector). One or more Tools services to query. Common service names include: `name2codes`. A validation step ensures that all provided services match the supported service names.<br>By default: all services. |

| | |
|---|---|
| params | `list`. A named list of query parameters to include in the request. The list of available parameters can be accessed via the EPPO API Documentation platform.<br>By default: no parameters, you need to specify the required ones. |
| apiKey | `character` (string). The API key used for authentication. It can be specified manually or through the .Renviron file.<br>By default: Sys.getenv("EPPO_API_KEY"). |

## Value

A named list of results, in which each level corresponds to the data retrieved for each specified service. Each service element contains a flattened dataframe with the queried content.

## Examples

```
## Not run:
# Get the EPPO codes associated to the name Bemisia tabaci.
toolsData_ <- tools(params = list(
  name2codes = list(
    onlyPreferred = FALSE,
    name = "Bemisia tabaci")))

## End(Not run)
```

---

`uniformTaxonomy` *Create a complete and uniform taxonomy dataframe.*

---

## Description

This function normalizes the taxonomy returned by the EPPO service, producing a uniform structure that includes all possible taxonomic categories, even when some of them are not present in the original result.

## Usage

```
uniformTaxonomy(taxonomyData)
```

## Arguments

| | |
|---|---|
| taxonomyData | `tibble`. A dataframe containing taxonomy data provided by the EPPO service for a given EPPO code. |

## Details

The function uses an internal set of known taxonomic ranks and performs a left join with the `taxonomy` dataframe, filling missing ranks with `NA`. After the join, the column `level` from the original taxonomy is removed, since the rank order is fully defined by the internal reference list. The column `queried_eppo_code` NA values are filled with the right code.

## Value

A dataframe where each row represents one of the expected taxonomic ranks. Fields corresponding to ranks not present in the original taxonomy are filled with `NA`. The `level` column is excluded from the output. The column `queried_eppo_code` NA values are filled with the right code.

## Examples

```
## Not run:
# Retrieve taxonomy data from the EPPO service.
taxonData_ <- taxon(eppoCodes = c("BEMITA"), services = c("taxonomy"))

# Create a uniform taxonomy with all ranks.
taxonomyData_ <- uniformTaxonomy(taxonomyData = taxonData_$taxonomy)

## End(Not run)
```

# Index

country, 2

general, 3

references, 4
reportingService, 5
rppo, 6

taxon, 8
taxons, 9
tools, 10

uniformTaxonomy, 11