Package: kvkapiR (via r-universe)

October 18, 2026

Title Interface to the Dutch Chamber of Commerce (KvK) API

Version 0.1.2

Description Access business registration data from the Dutch Chamber of Commerce (Kamer van Koophandel, KvK) through their official API <https://developers.kvk.nl/>. Search for companies by name, location, or registration number. Retrieve detailed business profiles, establishment information, and company name histories. Built on 'httr2' for robust API interaction with automatic pagination, error handling, and usage tracking.

License MIT + file LICENSE

URL https://coeneisma.github.io/kvkapiR/, https://github.com/coeneisma/kvkapiR/

BugReports https://github.com/coeneisma/kvkapiR/issues

Imports cli, dplyr, httr2, lifecycle, purrr, tibble, tidyr, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0), withr

Config/Needs/website rmarkdown

Config/testthat/edition 3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs libicu-dev libssl-dev

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

Date/Publication 2025-06-22 21:07:04 UTC

RemoteUrl https://github.com/coeneisma/kvkapir

RemoteRef HEAD

RemoteSha ffa81211dd4ad73fe8f4f481d61906a53fd80245

Contents

kvk_export_usage 2
kvk_get_basisprofiel 3
kvk_get_naamgeving 5
kvk_get_vestigingsprofiel 6
kvk_reset_usage 7
kvk_search 8
kvk_set_api_key 9
kvk_usage_alert 10
kvk_usage_report 11

Index **13**

`kvk_export_usage` *Export session usage data to CSV file*

Description

Export your KvK API usage data from the current session to a CSV file for external analysis, reporting, or record keeping. Choose between summary data or detailed call-by-call data.

Usage

```
kvk_export_usage(file_path, format = "summary")
```

Arguments

`file_path`	Character string specifying the path where the CSV file should be saved. Can include relative or absolute paths. The file will be created if it doesn't exist, or overwritten if it does.
`format`	Character string specifying the export format: • `"summary"` (default): Session summary with call counts by type and total costs. Ideal for quick reporting and budget tracking. • `"detailed"`: One row per API call with detailed information including timestamp, date, call type, and other metadata. Perfect for creating custom visualizations and detailed analysis.

Details

The exported data only includes production API calls from the current R session. Test environment calls are not included in the export. All data is session-based and will be lost when you restart R.

Summary format includes these columns:

- `Search`, `Basisprofiel`, `Vestiging`, `Naamgeving`: Call counts by type
- `Total Calls`: Total number of API calls in this session
- `Costs (EUR)`: Session costs (EUR 0.02 per paid call)

Detailed format includes these columns:

- `timestamp`: Exact time of the API call
- `date`: Date of the API call
- `year`, `month`: Extracted date components
- `call_type`: Type of API call (search, basisprofiel, etc.)
- `test_environment`: Always FALSE for production calls

Value

Invisibly returns NULL. The function is called for its side effect of creating a CSV file.

See Also

`kvk_usage_report` for viewing usage in R, `kvk_usage_alert` for setting usage limits, `kvk_reset_usage` for clearing usage history

Examples

```
# Export session summary (default)
kvk_export_usage("session_summary.csv")

# Export detailed call data for analysis
kvk_export_usage("detailed_calls.csv", format = "detailed")

# Export to specific directory
kvk_export_usage("reports/session_data.csv", format = "summary")
```

---

kvk_get_basisprofiel *Retrieve the Basisprofiel for a given KvK number*

Description

This function retrieves the basis profile for a given KvK number (`kvkNummer`) using the KvK Basisprofiel API. Users can also request additional data, such as owner information, main establishment, and other locations by specifying the `include` parameter.

Usage

```
kvk_get_basisprofiel(
  kvkNummer,
  geoData = FALSE,
  include = NULL,
  test_environment = FALSE
)
```

Arguments

kvkNummer	A string representing the KvK number for which the basis profile is requested.
geoData	A logical value indicating whether geo-data should be included in the response. Defaults to FALSE. If TRUE, the response will include geo-data.
include	A character vector specifying additional data to include. Possible values: "eigenaar" (owner), "hoofdvestiging" (main establishment), "vestigingen" (locations). Defaults to NULL (basic profile only).
test_environment	A logical value indicating whether to use the test environment (TRUE) or the production environment (FALSE). Defaults to FALSE.

Details

The function also supports the test_environment argument. When set to TRUE, the function will query the KvK test API environment, providing a set of fictitious test data.

The function retrieves data from the KvK Basisprofiel API. If geoData = TRUE, geo-data (e.g., location data) will be included in the returned profile. The include parameter allows users to request additional details, such as ownership information or business locations.

Important: If test_environment = TRUE, no actual API key is required. You will be using test data from the KvK test environment.

Value

A tibble containing the retrieved basis profile. If geoData = TRUE, the returned tibble will also include geographical information.

Examples

```
# Examples using the production API (requires API key)
# Retrieve basis profile for a given KvK number without geo-data
basis_profile <- kvk_get_basisprofiel(kvkNummer = "69599084")
print(basis_profile)

# Retrieve basis profile with geo-data
basis_profile_geo <- kvk_get_basisprofiel(kvkNummer = "69599084", geoData = TRUE)

# Retrieve basis profile including owner information
basis_profile_owner <- kvk_get_basisprofiel(kvkNummer = "69599084", include = "eigenaar")

# Examples using test environment (no API key required)
# These use the KvK test dataset with Donald Duck themed businesses
basis_profile_test <- kvk_get_basisprofiel(kvkNummer = "68727720", test_environment = TRUE)
print(basis_profile_test)
```

kvk_get_naamgeving *Retrieve the Naamgeving for a given KvK number*

Description

This function retrieves the name information (naamgeving) for a given KvK number (kvkNummer) using the KvK Naamgevingen API.

Usage

```
kvk_get_naamgeving(kvkNummer, test_environment = FALSE)
```

Arguments

kvkNummer	A string representing the KvK number for which the name information is requested.
test_environment	A logical value indicating whether to use the test environment (TRUE) or the production environment (FALSE). Defaults to FALSE.

Details

The function also supports the test_environment argument. When set to TRUE, the function will query the KvK test API environment, providing a set of fictitious test data.

The function retrieves data from the KvK Naamgevingen API.

Important: If test_environment = TRUE, no actual API key is required. You will be using test data from the KvK test environment.

Value

A tibble containing the retrieved name information.

Examples

```
# Examples using the production API (requires API key)
# Retrieve naamgeving for a given KvK number
naamgeving <- kvk_get_naamgeving(kvkNummer = "68750110")
print(naamgeving)

# Examples using test environment (no API key required)
naamgeving_test <- kvk_get_naamgeving(kvkNummer = "68727720", test_environment = TRUE)
print(naamgeving_test)
```

kvk_get_vestigingsprofiel *Retrieve the Vestigingsprofiel for a given establishment number*

Description

This function retrieves the establishment profile (`vestigingsprofiel`) for a given establishment number (`vestigingsnummer`) using the KvK Vestigingsprofiel API.

Usage

```
kvk_get_vestigingsprofiel(
  vestigingsnummer,
  geoData = FALSE,
  test_environment = FALSE
)
```

Arguments

`vestigingsnummer`	A string representing the establishment number for which the profile is requested.
`geoData`	A logical value indicating whether geo-data should be included in the response. Defaults to `FALSE`. If `TRUE`, the response will include geo-data.
`test_environment`	A logical value indicating whether to use the test environment (`TRUE`) or the production environment (`FALSE`). Defaults to `FALSE`.

Details

The function also supports the `test_environment` argument. When set to `TRUE`, the function will query the KvK test API environment, providing a set of fictitious test data.

The function retrieves data from the KvK Vestigingsprofiel API. If `geoData = TRUE`, geo-data (e.g., location data) will be included in the returned profile.

Important: If `test_environment = TRUE`, no actual API key is required. You will be using test data from the KvK test environment.

Value

A tibble containing the retrieved establishment profile. If `geoData = TRUE`, the returned tibble will also include geographical information.

Examples

```
# Examples using the production API (requires API key)
# Retrieve vestigingsprofiel for a given establishment number
vestigingsprofiel <- kvk_get_vestigingsprofiel(vestigingsnummer = "000038509504")
print(vestigingsprofiel)
```

```
# Retrieve vestigingsprofiel with geo-data
vestigingsprofiel_geo <- kvk_get_vestigingsprofiel(
  vestigingsnummer = "000038509504",
  geoData = TRUE
)

# Examples using test environment (no API key required)
vestigingsprofiel_test <- kvk_get_vestigingsprofiel(
  vestigingsnummer = "000019716893",
  test_environment = TRUE
)
print(vestigingsprofiel_test)
```

`kvk_reset_usage` *Reset session usage tracking data*

Description

Clear all API usage data from the current R session. This action cannot be undone and will reset your session usage history including call counts, timestamps, and cost calculations.

Usage

```
kvk_reset_usage()
```

Details

This function clears the session-based usage tracking data stored in memory. After reset:

- All session usage history is cleared
- `kvk_usage_report` will show no data
- Session cost calculations restart from zero
- Usage alerts continue with the new baseline

The function only affects session usage data. It does not change:

- Usage alert settings (these are stored in R options)
- API key configuration
- Package preferences

This can be useful for:

- Starting fresh tracking within a long R session
- Clearing test data before production use
- Resetting session counters for specific analyses
- Troubleshooting usage tracking issues

Value

Invisibly returns NULL. The function is called for its side effect of clearing session usage data.

See Also

kvk_usage_report for viewing current usage, kvk_export_usage for exporting session data, kvk_usage_alert for setting usage limits

Examples

```
# Clear session usage history
kvk_reset_usage()

# Verify reset worked
kvk_usage_report()  # Should show "No usage data found"

# Usage tracking will restart with new API calls
results <- kvk_search(plaats = "Amsterdam")
kvk_usage_report()  # Will show the new call
```

---

kvk_search *Retrieve all results from the KvK Search API (up to 1,000 results)*

Description

This function automatically paginates through the KvK API to collect available results. Due to API limitations, it retrieves a maximum of 1,000 records. When this happens, a warning will be displayed.

Usage

```
kvk_search(..., test_environment = FALSE)
```

Arguments

...	Named arguments passed to the API query (e.g., naam = "Koudum"). Available arguments can be found at https://developers.kvk.nl/documentation/zoeken-api#input.
test_environment	A logical value. If TRUE, uses the test environment instead of the live API. Defaults to FALSE.

Details

If test_environment = TRUE, it switches to the KvK's test environment, using a mock API key for testing purposes.

Value

A tibble containing the retrieved results. Possible parameters can be found under the `Results` section of https://developers.kvk.nl/documentation/zoeken-api#output.

Examples

```
# Examples using the production API (requires API key)
koudum <- kvk_search(plaats = "Koudum")
print(koudum)

rotterdam <- kvk_search(plaats = "Rotterdam")
print(rotterdam)

# Examples using test environment (no API key required)
test_data <- kvk_search(plaats = "Utrecht", test_environment = TRUE)
print(test_data)
```

---

kvk_set_api_key *Set KvK API Key for the current session*

Description

This function sets the specified KvK API key for the **current R session** using `Sys.setenv()`. The key will be available until the session ends.

Usage

```
kvk_set_api_key(api_key)
```

Arguments

`api_key`	A string containing the API key.

Details

The function sets the API key using `Sys.setenv()`, making it available for the current session only.

To store your API key permanently across R sessions, add the following line to your .Renviron file:

`KVK_API_KEY=l7xxYourActualApiKeyHere1234567890abcdef`

You can edit your .Renviron file by running: `usethis::edit_r_environ()`

After adding the key, restart R for the changes to take effect.

Value

Invisibly returns `TRUE` if the key was set.

Examples

```
# Set the API key for the current session
kvk_set_api_key("abcd1234")

# To store permanently, add this line to .Renviron:
# KVK_API_KEY=your_actual_api_key_here
```

`kvk_usage_alert` *Set usage and cost alerts for current session*

Description

Configure automatic alerts to monitor your KvK API usage within the current R session. Alerts are triggered in real-time when you exceed the specified limits during API calls.

Usage

```
kvk_usage_alert(max_calls = NULL, max_cost = NULL)
```

Arguments

`max_calls`	Integer maximum number of paid API calls before triggering alerts, or NULL to disable call alerts. Only basisprofiel, vestigingsprofiel, and naamgeving calls count towards this limit. Search calls are free and do not count. When reached, you'll see a danger alert. Continued usage beyond this limit will show reminder warnings.
`max_cost`	Numeric maximum cost in euros before triggering alerts, or NULL to disable cost alerts. For example, 5.45 represents EUR 5.45. Costs are calculated as EUR 0.02 per paid API call (search calls are free).

Details

Usage alerts help you control API consumption within your current R session. When limits are exceeded, the package displays informative messages:

- **First alert**: Danger alert when the limit is first reached
- **Reminder alerts**: Warning messages for continued usage over limits

Alert settings are stored in R options and apply only to the current session. All alert configurations are automatically reset when you restart R. To disable all alerts within the current session, call the function with both parameters set to NULL: `kvk_usage_alert()`.

Only production API calls count towards limits. Test environment calls are ignored.

Value

No return value, called for side effects. Sets usage alert options for the current session.

See Also

kvk_usage_report for viewing current usage, kvk_export_usage for exporting usage data, kvk_reset_usage for clearing usage history

Examples

```
# Set session limit for paid calls only (search calls don't count)
kvk_usage_alert(max_calls = 100)

# Set session cost limit to EUR 5.00
kvk_usage_alert(max_cost = 5.00)

# Set both paid call and cost limits (EUR 10.50)
kvk_usage_alert(max_calls = 500, max_cost = 10.50)

# Disable all alerts for this session
kvk_usage_alert()
```

---

kvk_usage_report *Generate API usage report for current session*

Description

Display or return API usage statistics and costs for the current R session. This function helps you monitor your KvK API usage and associated costs.

Usage

```
kvk_usage_report(format = "summary")
```

Arguments

format	Character string specifying output format: • "summary" (default): Human-readable console output with costs and call counts for this session • "tibble": Structured summary data as a tibble • "detailed": Detailed data with one row per API call, suitable for analysis and visualization

Details

The summary format displays API call counts and costs for the current R session only. All usage data is session-based and will be reset when you restart R. Search calls are free, while profile calls cost EUR 0.02 each.

Only production API calls are included in reports. Test environment calls are not tracked.

Value

- `format = "summary"`: Invisible NULL (output printed to console)
- `format = "tibble"`: A tibble with session usage summary
- `format = "detailed"`: A tibble with detailed call-by-call data

See Also

`kvk_usage_alert` for setting usage limits, `kvk_export_usage` for exporting data to CSV, `kvk_reset_usage` for clearing usage history

Examples

```
# Display human-readable summary
kvk_usage_report()

# Get structured summary as tibble
summary_data <- kvk_usage_report(format = "tibble")
print(summary_data)

# Get detailed call data for custom analysis
call_data <- kvk_usage_report(format = "detailed")
head(call_data)
```

Index

kvk_export_usage, 2, *8*, *11, 12*
kvk_get_basisprofiel, 3
kvk_get_naamgeving, 5
kvk_get_vestigingsprofiel, 6
kvk_reset_usage, *3*, 7, *11, 12*
kvk_search, 8
kvk_set_api_key, 9
kvk_usage_alert, *3*, *8*, 10, *12*
kvk_usage_report, *3*, *7, 8*, 11, *11*