Package: genderapi (via r-universe)

October 30, 2026

Type Package

Title Official 'GenderAPI.io' V2 Client

Version 2.0.0

Description Official 'GenderAPI.io' V2 client for R. Provides an interface to the 'GenderAPI.io' V2 web service <https://www.genderapi.io/api-documentation> that infers gender from personal names, email addresses and usernames, runs batches of up to 50 items, reads credit usage and validates phone numbers. Responses are returned as parsed lists with all fields kept, including unknown results, confidence metadata, billing status and batch summaries; errors are raised as structured conditions. Requests are never retried and redirects are never followed. Results are inferences, not verified identity, and can be unknown.

License MIT + file LICENSE

URL https://www.genderapi.io/api-documentation, https://github.com/GenderAPI/genderapi-R

BugReports https://github.com/GenderAPI/genderapi-R/issues

Depends R (>= 4.1.0)

Imports curl (>= 5.0.0), jsonlite, utils

Suggests testthat (>= 3.0.0), webfakes

Config/testthat/edition 3

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Config/pak/sysreqs libssl-dev

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

Date/Publication 2026-09-30 15:18:43 UTC

RemoteUrl https://github.com/genderapi/genderapi-r

RemoteRef HEAD

RemoteSha a9c1397b98ef47e99f3769134a765108c5ef91bf

Contents

as.data.frame.genderapi 2
genderapi_batch 3
genderapi_capabilities 4
genderapi_client 5
genderapi_error 6
genderapi_failed 7
genderapi_gender 8
genderapi_item 10
genderapi_usage 11
genderapi_validate_phone 12

Index **13**

`as.data.frame.genderapi` *Convert GenderAPI results to a data frame*

Description

Flattens a prediction (one row) or a batch (one row per item, in submission order) into a data frame. JSON `null` becomes `NA`. `confidence` is copied unchanged; it is not converted into a percentage or probability. The original list keeps every field, including fields not shown here.

Usage

```
## S3 method for class 'genderapi_prediction'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'genderapi_batch'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

`x`	A `genderapi_prediction` or `genderapi_batch`.
`row.names`, `optional`	Ignored; present for compatibility with the generic.
`...`	Ignored.

Details

Columns: `index`, `id`, `charged_credits` (batch only), `input_type`, `input_value`, `gender`, `result_status`, `reason`, `confidence`, `confidence_kind`, `sample_count`, `source`, `name`, `country`, `country_source`, `match_name`, `match_method`, `match_scope`, `match_country`, and for batches `error_status`, `error_code`, `error_detail`.

Value

A data frame.

Examples

```
res <- structure(
  list(data = list(gender = NULL, result_status = "unknown",
                   reason = "not_found", confidence = NULL,
                   confidence_kind = NULL, source = "none"),
       meta = list()),
  class = c("genderapi_prediction", "genderapi_response")
)
as.data.frame(res)
```

---

genderapi_batch *Infer gender for a batch of 1 to 50 items*

Description

Sends one `POST /gender/batch` request with `{"items": [...]}`. Batch items default to `ai_mode = "off"` on the server. The server decides the limit (50 items; 10 on the IP trial). Larger jobs must be split by the caller; the package never retries or splits automatically.

Usage

```
genderapi_batch(items, client = genderapi_client())
```

Arguments

`items`	The items: a list of `genderapi_item()` objects, a list of plain lists using either the argument names of `genderapi_item()` or the wire names (`type`, `value`, `country`, `id`, `forceToGenderize`, `options = list(ai_mode = ...)`), or a data frame with columns `type`, `value` and optionally `country`, `ai_mode`, `force_to_genderize`, `id` (`NA` means "not set").
`client`	A `genderapi_client()`.

Details

A partially successful batch is returned normally (HTTP 200): inspect every item. Each item has `index`, the optional `id`, `charged_credits` and exactly one of `data` (a prediction) or `error` (a problem with `code`). `meta$summary` has `total`, `succeeded`, `identified`, `unknown` and `failed`. When every executed item fails the API answers with an error status; the resulting `genderapi_http_error` keeps the item array in `data`.

Retry only failed items, and only once billing is confirmed: resubmitting successful items charges them again.

Value

A `genderapi_batch`: the parsed V2 JSON with `data` (the item list) and `meta` (including `summary` and `usage`). Use `as.data.frame()` for one row per item and `genderapi_failed()` for the failed items.

Examples

```
items <- list(
  genderapi_item("name", "Onur", country = "TR", id = "a"),
  genderapi_item("email", "alex@example.com", id = "b")
)
## Not run:
# Sends a request and may consume credits.
res <- genderapi_batch(items)
res$meta$summary
as.data.frame(res)
genderapi_failed(res)

## End(Not run)
```

`genderapi_capabilities` *Read API capabilities and the error catalog*

Description

`genderapi_capabilities()` sends `GET /` (deployment version, limits and AI availability). `genderapi_error_catalog()` sends `GET /errors`, the public catalog of stable error codes, HTTP statuses and recommended actions. Both are unauthenticated: no API key is sent.

Usage

```
genderapi_capabilities(client = genderapi_client())

genderapi_error_catalog(client = genderapi_client())
```

Arguments

client	A genderapi_client().

Value

The parsed JSON object as a list of class genderapi_response.

Examples

```
## Not run:
genderapi_capabilities()
catalog <- genderapi_error_catalog()

## End(Not run)
```

---

genderapi_client *Create a GenderAPI.io V2 client*

Description

A client holds the API key, base URL, timeout and user agent. Creating a client never sends a request. Every request function takes a client argument that defaults to genderapi_client(), so setting the GENDERAPI_API_KEY environment variable is enough for most scripts.

Usage

```
genderapi_client(
  api_key = Sys.getenv("GENDERAPI_API_KEY", ""),
  base_url = default_base_url,
  timeout = 10,
  user_agent = NULL,
  require_api_key_access = TRUE
)
```

Arguments

api_key	API key string. Defaults to the GENDERAPI_API_KEY environment variable. NULL or "" means no key (IP trial).
base_url	API base URL. Keep the default in production. HTTPS is required; http:// is accepted only for localhost, 127.0.0.1 and [::1] so that tests can use a local fake server.
timeout	Total request timeout in seconds (default 10). A timed-out request is not retried and may still have been billed.
user_agent	Optional User-Agent header. Defaults to genderapi-r/<version>.

require_api_key_access
: TRUE (default) to raise a genderapi_access_mode_error when a key is set but a successful response reports an access mode other than "api_key". Applies to predictions, batches, usage and phone validation, never to genderapi_capabilities() or genderapi_error_catalog(). Has no effect without a key. See genderapi_error.

Details

Without a key (api_key is NULL or empty and GENDERAPI_API_KEY is unset) requests are sent without an Authorization header and the server applies its shared IP trial (10 credits per 24 hours per public IP). The package has no client-side trial logic; meta$access$mode in each response tells you which access mode the server used.

When a key is set, the package by default checks that each successful authenticated response reports meta$access$mode == "api_key". If the server answered through another mode (usually "ip_trial" because the key was not recognized), a genderapi_access_mode_error is raised. The request has already been processed and may have consumed IP-trial credits; the full result is in the error's result field. It is never retried. Set require_api_key_access = FALSE to return such responses normally.

Keep keys server-side. Never put a key in browser code, a URL, a log or a document shared with end users. Printing a client never shows the key.

Value

An object of class genderapi_client.

Examples

```
# No request is sent here.
client <- genderapi_client(api_key = NULL, timeout = 5)
client
```

---

genderapi_error *GenderAPI error conditions*

Description

All errors raised by this package inherit from genderapi_error, so they can be caught with tryCatch(..., genderapi_error = function(e) ...). More specific classes:

Details

- genderapi_validation_error: the input failed a client-side check. No request was sent and nothing was billed.

- genderapi_http_error: the API answered with HTTP 400 or higher. Fields: status, code, title, detail, action, errors (validation pointers), documentation, request_id (from the body, meta$request_id or the X-Request-ID header), retry_after (the Retry-After header as a string), billing_status, usage (meta$usage), meta, data (the item array of an all-failed batch), body (parsed JSON or NULL), raw (the response text) and headers.
- genderapi_redirect_error: the API answered with a 3xx redirect. It was not followed, so the key was not forwarded. Fields: status, location, headers.
- genderapi_transport_error: no usable response (network error or timeout). code is "timeout" or "transport_error". The request may still have been processed and billed.
- genderapi_response_error: a 2xx response that is not a JSON object. code is "invalid_response".
- genderapi_access_mode_error: an API key is set but a successful response reports an access mode other than "api_key" in meta$access$mode (usually "ip_trial", because the key was not recognized). code is "unexpected_access_mode". The request has already been processed and may have consumed IP-trial credits; it is not retried. Fields: access_mode, access_reason, result (the complete parsed result the function would have returned, including meta$usage), status and request_id. Disable the check with genderapi_client(require_api_key_access = FALSE).

The package never retries. After a 429 wait for retry_after seconds; the next request is a new, billable operation. When billing_status is "unconfirmed" or action is "contact_support", contact support with request_id before sending the request again. Match on code, never on the human-readable detail. See genderapi_error_catalog().

Error bodies can contain the submitted input. Do not log them wholesale.

Examples

```
client <- genderapi_client(api_key = NULL)
e <- tryCatch(genderapi_name("", client = client), genderapi_error = function(e) e)
class(e)
conditionMessage(e)
```

---

genderapi_failed *Failed items of a batch*

Description

Returns the items that carry an error instead of data. Works on a genderapi_batch (partial success) and on the genderapi_http_error raised when every executed item failed. Retry failed items only after billing is confirmed, and never resubmit successful items: they would be charged again.

Usage

```
genderapi_failed(x)
```

Arguments

x	A genderapi_batch or a genderapi_http_error from genderapi_batch().

Value

A list of items, each with `index`, optional `id`, `charged_credits` and `error` (a problem with `code`, `status`, `detail` and `action`). An empty list when nothing failed.

Examples

```
res <- structure(
  list(data = list(
    list(index = 0L, id = "a", charged_credits = 1L,
         data = list(gender = "male", result_status = "identified")),
    list(index = 1L, id = "b", charged_credits = 0L,
         error = list(code = "ai_upstream_error", status = 502L))
  ), meta = list()),
  class = c("genderapi_batch", "genderapi_response")
)
genderapi_failed(res)
```

`genderapi_gender` *Infer gender for one name, email address or username*

Description

Sends one `POST /gender` request (one billable operation). The request is never retried. A successful unknown result is billable and is returned, not raised as an error.

Usage

```
genderapi_gender(
  type,
  value,
  country = NULL,
  ai_mode = NULL,
  force_to_genderize = FALSE,
  id = NULL,
  client = genderapi_client()
)

genderapi_name(
  value,
  country = NULL,
  ai_mode = NULL,
  force_to_genderize = FALSE,
  id = NULL,
  client = genderapi_client()
)

genderapi_email(
```

```
  value,
  country = NULL,
  ai_mode = NULL,
  force_to_genderize = FALSE,
  id = NULL,
  client = genderapi_client()
)

genderapi_username(
  value,
  country = NULL,
  ai_mode = NULL,
  force_to_genderize = FALSE,
  id = NULL,
  client = genderapi_client()
)
```

Arguments

type	One of "name", "email" or "username".
value	The name, email address or username: 1 to 254 characters, not only whitespace, no control characters.
country	Optional ISO 3166-1 alpha-2 code in upper case, such as "TR". Omit it when unknown.
ai_mode	Optional "off", "fallback" or "always". When omitted the server default applies: fallback for single requests and off for batch items. fallback costs 1 credit in total; always costs 2.
force_to_genderize	TRUE to try the dataset first and, if gender is unknown, infer it from nickname or alias semantics (2 credits in total in that case). Cannot be combined with ai_mode "off" or "always".
id	Optional item id, 1 to 64 characters. In a batch, ids must be unique and are echoed back in each result.
client	A genderapi_client().

Details

genderapi_name(), genderapi_email() and genderapi_username() are shortcuts for genderapi_gender() with the matching type.

Value

A genderapi_prediction: the parsed V2 JSON as a list with data (the prediction) and meta (request_id, duration_ms, access, usage). All fields are kept as sent, including fields added by future API versions. JSON null becomes NULL. Key fields of data: gender ("male", "female" or NULL), result_status ("identified" or "unknown"), reason, confidence with confidence_kind ("observed_frequency" or "model_reported"; not a calibrated probability), sample_count,

source, name, country, country_source and match. Use as.data.frame() for a one-row data frame.

See Also

genderapi_batch(), genderapi_error

Examples

```
## Not run:
# Sends a request and may consume credits.
res <- genderapi_name("Onur", country = "TR")
res$data$gender
res$data$result_status
res$meta$usage$charged_credits
as.data.frame(res)

## End(Not run)
```

---

genderapi_item *Build one prediction item*

Description

Validates the input with the cheap, certain rules of the V2 schema and returns the exact wire object (type, value, country, id, forceToGenderize, options$ai_mode). Invalid input raises a genderapi_validation_error; no request is sent. The API performs the authoritative checks (email syntax, country membership, trial limits).

Usage

```
genderapi_item(
  type,
  value,
  country = NULL,
  ai_mode = NULL,
  force_to_genderize = FALSE,
  id = NULL
)
```

Arguments

type	One of "name", "email" or "username".
value	The name, email address or username: 1 to 254 characters, not only whitespace, no control characters.
country	Optional ISO 3166-1 alpha-2 code in upper case, such as "TR". Omit it when unknown.

ai_mode	Optional "off", "fallback" or "always". When omitted the server default applies: fallback for single requests and off for batch items. fallback costs 1 credit in total; always costs 2.
force_to_genderize	TRUE to try the dataset first and, if gender is unknown, infer it from nickname or alias semantics (2 credits in total in that case). Cannot be combined with ai_mode "off" or "always".
id	Optional item id, 1 to 64 characters. In a batch, ids must be unique and are echoed back in each result.

Value

A list of class genderapi_item.

Examples

```
genderapi_item("name", "Onur", country = "TR", ai_mode = "off", id = "row-1")
```

---

genderapi_usage *Read credit usage (free)*

Description

Sends GET /usage. This read is not billed. data holds remaining_credits (can be negative or NULL), expires_at, and for the IP trial resets_at, limit and period_seconds; meta$access$mode shows whether the key or the IP trial was used.

Usage

```
genderapi_usage(client = genderapi_client())
```

Arguments

client	A genderapi_client().

Value

A genderapi_usage list with data and meta.

Examples

```
## Not run:
u <- genderapi_usage()
u$data$remaining_credits
u$meta$access$mode

## End(Not run)
```

genderapi_validate_phone *Validate a phone number*

Description

Sends one POST /phone/validate request. It costs 1 credit, including for invalid numbers, and is never retried.

Usage

```
genderapi_validate_phone(number, country = NULL, client = genderapi_client())
```

Arguments

number	Phone number, 3 to 32 characters of digits, spaces, parentheses and hyphens with an optional leading +.
country	Optional upper-case ISO 3166-1 alpha-2 code used for numbers without an international prefix.
client	A genderapi_client().

Value

A genderapi_phone list. data has valid, possible, e164, country and country_calling_code.

Examples

```
## Not run:
genderapi_validate_phone("+90 212 555 01 01")$data$valid

## End(Not run)
```

Index

as.data.frame(), *4*, *10*
as.data.frame.genderapi, 2
as.data.frame.genderapi_batch
        *(*as.data.frame.genderapi*)*, 2
as.data.frame.genderapi_prediction
        *(*as.data.frame.genderapi*)*, 2

genderapi_batch, 3
genderapi_batch(), *7*, *10*
genderapi_capabilities, 4
genderapi_capabilities(), *6*
genderapi_client, 5
genderapi_client(), *3*, *5*, *9*, *11, 12*
genderapi_email *(*genderapi_gender*)*, 8
genderapi_error, 6, *6*, *10*
genderapi_error_catalog
        *(*genderapi_capabilities*)*, 4
genderapi_error_catalog(), *6, 7*
genderapi_failed, 7
genderapi_failed(), *4*
genderapi_gender, 8
genderapi_item, 10
genderapi_item(), *3*
genderapi_name *(*genderapi_gender*)*, 8
genderapi_usage, 11
genderapi_username *(*genderapi_gender*)*, 8
genderapi_validate_phone, 12