

Package ‘pacha’

September 24, 2026

Title Reproducible Reporting for ChecklistBank Ethnobotanical Data

Version 0.1.1

Description Provides programmatic access to the 'Listado de plantas de uso y aprovechamiento sostenible en Ecuador' <[doi:10.48580/dgvrn](https://doi.org/10.48580/dgvrn)>. Acts as a client for the 'ChecklistBank' API and for processing 'ColDP' (Catalogue of Life Data Package) files. Supports data retrieval for ethnobotanical records and includes built-in Markdown report generation for reproducible workflows.

License GPL (>= 3)

Language en-US

URL <https://github.com/envinatu/pacha>

BugReports <https://github.com/envinatu/pacha/issues>

Imports httr2, stats, tools, utils, yaml

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

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

NeedsCompilation no

Author Marcelo F. Amores-Palma [aut, cre, cph] (ORCID: <<https://orcid.org/0000-0002-0337-0467>>)

Maintainer Marcelo F. Amores-Palma <contacto.envinatu@gmail.com>

Repository CRAN

Date/Publication 2026-09-24 14:00:02 UTC

Contents

accessors-pacha	2
common_names_pacha	3

compare_pacha	4
establishment_pacha	5
indexation_urls_pacha	6
is_listed_pacha	7
pacha_clear_cache	8
pacha_configure	9
pacha_report	11
pacha_sc_full_name	12
reference_pacha	14
sustainable_uses_pacha	15
threat_status_pacha	16
Index	18

accessors-pacha	<i>Plain-text and Markdown accessors</i>
-----------------	--

Description

The package functions generally follow a dual-interface design: plain-text variants (*_pacha()) for interactive exploration, and Markdown variants (*_pacha_md()) for document rendering. Both share the same underlying source logic, differing only in formatting and error handling. Specific function availability is detailed in their respective documentation pages.

Plain-text accessors (*_pacha())

Designed for interactive lookups and inline queries about a single taxon. They prioritize explicit feedback:

- Return a localized message when data is missing.
- Raise an R warning() and return a connection message on network failure.
- Warn when ColDP taxon resolution requires disambiguation.
- Print output directly to the console via cat() (returning invisibly) to preserve multi-line formatting (except indexation_urls_pacha()).

Markdown accessors (*_pacha_md())

Designed for assembling multi-component or multi-taxon Markdown reports. They prioritize clean document compilation:

- Suppress all R warning()s, including connection and disambiguation issues.
- Return an empty string ("") invisibly when data is unavailable, preventing broken layouts or redundant headers in the compiled document.

Choosing between them

Use *_pacha() when inspecting an individual taxon interactively. Use *_pacha_md() when generating dynamic reports (e.g., Quarto or R Markdown), where unavailable data should omit silently rather than disrupt document rendering.

common_names_pacha	<i>Retrieve common names</i>
--------------------	------------------------------

Description

Retrieves vernacular names for a scientific name from the currently configured source. The API source queries the configured ChecklistBank dataset; the local source reads VernacularName.tsv from the configured ColDP archive and, when a country column is present, keeps only the entries matching coldp_country. Names are grouped by language in the returned value. The examples below use the Ecuadorian checklist dataset configured by default, but the same calls work against any compatible ChecklistBank dataset or ColDP archive.

Usage

```
common_names_pacha(species, language = NULL, source = NULL, refresh = FALSE)
```

```
common_names_pacha_md(species, language = NULL, source = NULL, refresh = FALSE)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
language	Optional output-language code or alias used for labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.

Value

common_names_pacha() prints and invisibly returns a single character string of names grouped by language, or the localized no-data/connection message. common_names_pacha_md() returns, invisibly, the emitted Markdown string, or "" when no names are available.

See Also

[accessors-pacha](#) for how plain-text and Markdown accessors work.

Examples

```
## Not run:
common_names_pacha("Bidens andicola")
common_names_pacha("Bidens andicola", source = "coldp")

## End(Not run)
## Not run:
common_names_pacha_md("Bidens andicola")

## End(Not run)
```

compare_pacha

Compare the web API and local ColDP source

Description

Validates and compares the data returned by both sources for one species. The comparison always queries the API and the local ColDP archive, regardless of the configured default source. It normalizes case, accents, punctuation, and whitespace before assessing agreement, while retaining the original source values in the printed result and returned report. Sustainable uses are compared as category :: value pairs, so a value moved between categories is reported as a discrepancy rather than overlooked.

Usage

```
compare_pacha(
  species,
  component = "all",
  language = NULL,
  refresh = FALSE,
  print = TRUE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
component	One or more components to compare. Use "all" (the default) for the full comparison, or select "common_names", "sustainable_uses", "indexation_urls", "establishment", and/or "threat_status" individually. Recognized aliases map to these standard components as follows: "names" to "common_names"; "uses" to "sustainable_uses"; "urls" to "indexation_urls"; "origin" to "establishment"; and "threat" and "conservation" to "threat_status". Matching is case-insensitive and tolerant of spaces or hyphens in place of underscores.
language	Optional output-language code or alias retained in the report metadata; the comparison itself is language-neutral.
refresh	Logical scalar. If TRUE, forces both source records to be refreshed before comparison.
print	Logical scalar. If TRUE, prints the comparison result.

Details

The function prints a concise, component-by-component result by default and returns the full machine-readable report invisibly. Status values are "identical", "partial_overlap", "discrepant", "one_source_missing", "both_missing", "api_unavailable", and "coldp_unavailable". With component = "all", it compares common names, sustainable uses, indexation URLs, establishment status, and threat status.

Value

Invisibly, a list with species, language, components, summary, details, api_record, coldp_record, and compared_at. summary is a data frame with one row per selected component and its agreement status.

Examples

```
## Not run:
compare_pacha("Bidens andicola")
compare_pacha("Bidens andicola", component = "sustainable_uses")
compare_pacha("Bidens andicola", component = c("origin", "threat"))
compare_pacha("Bidens andicola", component = c("names", "urls"), print = FALSE)

## End(Not run)
```

establishment_pacha	<i>Retrieve establishment status</i>
---------------------	--------------------------------------

Description

Retrieves the documented establishment means of a taxon – native, introduced, cultivated, naturalized, and so on. The API source reads the ChecklistBank taxon-distribution endpoint. The local source reads Distribution.tsv and prefers the entry matching the configured coldp_country; when several equally relevant records remain, it selects the lowest ColDP identifier and reports this through a warning raised by the plain-text accessor.

Usage

```
establishment_pacha(species, language = NULL, source = NULL, refresh = FALSE)

establishment_pacha_md(
  species,
  language = NULL,
  source = NULL,
  refresh = FALSE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
language	Optional output-language code or alias used to translate the establishment category and labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.

Details

Values are translated through the selected language dictionary's establishment mapping. Values absent from that mapping are returned in sentence case rather than rejected, so datasets using additional establishment categories remain usable.

Value

`establishment_pacha()` prints and invisibly returns a translated establishment category, or the localized no-data/connection message. `establishment_pacha_md()` returns, invisibly, the emitted Markdown string, or "" when the source has no establishment record.

See Also

[accessors-pacha](#) for a detailed description of plain-text and Markdown accessors.

Examples

```
## Not run:
establishment_pacha("Bidens andicola")
establishment_pacha("Bidens andicola", source = "coldp", language = "en")

## End(Not run)
## Not run:
establishment_pacha_md("Bidens andicola", language = "en")

## End(Not run)
```

indexation_urls_pacha *Retrieve indexation URLs*

Description

Retrieves URLs associated with the resolved taxon. For the web source, the result includes any URLs returned by ChecklistBank plus its canonical taxon page. For the local source, URLs are extracted from NameUsage.tsv; IPNI links are normalized to their canonical `https://www.ipni.org/n/...` form.

Usage

```
indexation_urls_pacha(species, language = NULL, source = NULL, refresh = FALSE)

indexation_urls_pacha_md(
  species,
  language = NULL,
  source = NULL,
  refresh = FALSE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
language	Optional output-language code or alias used for labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.

Value

indexation_urls_pacha() returns, visibly, a named character vector of URLs, or the localized no-data/connection message. indexation_urls_pacha_md() returns, invisibly, the emitted Markdown string, or "" when no URLs are available.

See Also

[accessors-pacha](#) for a detailed description of plain-text and Markdown accessors.

Examples

```
## Not run:
indexation_urls_pacha("Bidens andicola")
indexation_urls_pacha("Bidens andicola", source = "coldp")

## End(Not run)
## Not run:
indexation_urls_pacha_md("Bidens andicola")

## End(Not run)
```

is_listed_pacha

Report whether a species is listed in the configured dataset

Description

Resolves a scientific name against the currently configured source (api or coldp) and reports, as a plain-text console message, whether the species is listed in the working dataset. The message is taken from the active language configuration and can be customized through the package configuration; if the required messages are unavailable for the selected language, the function raises an error. The language argument is resolved in the same way as for the other accessors in this module, accepting either a language alias or a language code. As with those accessors, a taxonomic match resolved from the local ColDP archive that required disambiguation raises one or more additional warning()s.

Usage

```
is_listed_pacha(
  species,
  language = NULL,
  source = NULL,
  refresh = FALSE,
  detailed = FALSE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
language	Optional output-language code or alias used for labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.
detailed	Logical scalar. If TRUE and the species is listed, the concrete sustainable-use records are appended, one category per block and one use per line, in plain text without Markdown formatting.

Value

Invisibly, the printed plain-text statement: the species name, the localized listed/not-listed statement, and, when detailed = TRUE and the species is listed, its sustainable uses.

Examples

```
## Not run:
is_listed_pacha("Bidens andicola")
is_listed_pacha("Bidens andicola", detailed = TRUE)
is_listed_pacha("Bidens andicola", language = "en", source = "coldp")

## End(Not run)
```

pacha_clear_cache	<i>Clear cached source records</i>
-------------------	------------------------------------

Description

Removes cached records for both sources. With species = NULL, also discards the cached ColDP tables, so subsequent local queries reread the ZIP archive. Supplying a species removes only that species from the API and ColDP record caches.

Usage

```
pacha_clear_cache(species = NULL)
```

Arguments

species Optional scientific name containing at least a genus and specific epithet. NULL clears every record and local-table cache.

Value

Invisibly, NULL.

Examples

```
pacha_clear_cache()
pacha_clear_cache("Bidens andicola")
```

pacha_configure	<i>Configure the pacha data module</i>
-----------------	--

Description

Gets or updates the configuration shared by every public accessor in this module. The accessors are source-agnostic: they query any ChecklistBank-compatible web API, or read any compatible local Catalogue of Life Data Package (ColDP) archive, provided the source exposes name usages and, where relevant, vernacular names, taxon properties, and distributions. The defaults set here – the ChecklistBank web API as the active source, dataset "313479" (the Ecuadorian checklist of plants susceptible to sustainable use), and coldp_country = "EC" – identify the dataset used in this file's \dontrun examples; any other ChecklistBank dataset or compatible ColDP archive can be configured in its place without modifying the module itself.

Usage

```
pacha_configure(
  dataset = NULL,
  base_url = NULL,
  source_name = NULL,
  language = NULL,
  fetcher,
  use_exclude_pattern = NULL,
  labels = NULL,
  source = NULL,
  coldp_zip_file = NULL,
  coldp_country = NULL,
  timeout = NULL
)
```

Arguments

dataset	Character scalar. ChecklistBank dataset identifier used by the default API adapter.
base_url	Character scalar. HTTP(S) base URL of the ChecklistBank API.
source_name	Character scalar identifying the web source in connection warnings.
language	Character scalar. A language-dictionary file stem or one of its aliases. Controls labels and presentation, not the source data.
fetcher	Optional custom web adapter. Pass NULL explicitly to restore the built-in ChecklistBank adapter; omit this argument to leave the current adapter unchanged.
use_exclude_pattern	Character scalar regular expression identifying taxon-property names that must not be treated as sustainable-use classes.
labels	Optional named list of named character vectors overriding the default labels by language.
source	Default source: "api"/"web" or "coldp"/"local".
coldp_zip_file	Path to a local ColDP ZIP archive. It must contain NameUsage.tsv; VernacularName.tsv, TaxonProperty.tsv, and Distribution.tsv are read when present. NULL leaves the configured path unchanged. The pacha_COLDP_ZIP environment variable is used lazily when no path is set.
coldp_country	ISO country code used to select country-specific records from the local ColDP archive: vernacular names in VernacularName.tsv, and the preferred distribution row in Distribution.tsv for establishment and threat status. Defaults to "EC".
timeout	Positive numeric HTTP timeout, in seconds, for web requests.

Details

fetcher preserves the existing extension contract for the web source. It receives (species, config) and must return a list with optional common_names, sustainable_uses, indexation_urls, establishment, threat_status, transport_error, and transport_messages elements. A custom fetcher applies only when source = "api"; source = "coldp" always reads the archive. For example:

```
my_fetcher <- function(species, config) {
  list(
    common_names = list(es = c("nombre comun")),
    sustainable_uses = list(Medicinal = c("uso medicinal registrado")),
    indexation_urls = c(source = "https://example.org/taxon/123"),
    establishment = "native",
    threat_status = "LC",
    transport_error = FALSE,
    transport_messages = character()
  )
}
pacha_configure(fetcher = my_fetcher)
```

Value

A named configuration list, returned visibly when called with no arguments, and invisibly after an update.

Examples

```
## Not run:
pacha_configure()
pacha_configure(source = "coldp", coldp_zip_file = "C:/data/ColdP.zip")
pacha_configure(source = "api", dataset = "313479", language = "es")

## End(Not run)
```

pacha_report	<i>Generate a per-species Markdown report</i>
--------------	---

Description

Concatenates the Markdown output of [pacha_sc_full_name_md\(\)](#), [common_names_pacha_md\(\)](#), and [sustainable_uses_pacha_md\(\)](#) into a single Markdown-formatted string, with one block per species in species, suitable for a results = "asis" knitr chunk in an automated report.

Usage

```
pacha_report(
  species,
  language = NULL,
  source = NULL,
  use = NULL,
  refresh = FALSE,
  print = TRUE
)
```

Arguments

species	Character vector of one or more scientific names, each containing at least a genus and a specific epithet.
language	Optional output-language code or alias, forwarded to common_names_pacha_md() and sustainable_uses_pacha_md() .
source	Optional source override ("api"/"web" or "coldp"/"local"), forwarded to the same two functions. NULL uses each function's configured default.
use	Optional sustainable-use category filter, forwarded to sustainable_uses_pacha_md() . NULL includes every category.
refresh	Logical scalar. If TRUE, bypasses the cache of common_names_pacha_md() and sustainable_uses_pacha_md() and re-fetches those two components for every species.

`print` Logical scalar. If TRUE (the default), the combined Markdown is emitted via `cat()`, as required in a `results = "asis"` knitr chunk. If FALSE, the string is built and returned without being printed.

Details

The heading for each species always comes from `pacha_sc_full_name_md()`, which takes no argument beyond species and resolves against its own dataset and cache, independently of language and source; see `?pacha_sc_full_name_md`. `language` and `source` are forwarded to both `common_names_pacha_md()` and `sustainable_uses_pacha_md()`; `use` is forwarded only to `sustainable_uses_pacha_md()`; `refresh` is forwarded to both `common_names_pacha_md()` and `sustainable_uses_pacha_md()`, but not to `pacha_sc_full_name_md()`, as scientific names rely on a separate cache unaffected by this parameter.

Each component is produced by capturing, not re-implementing, the output of the corresponding `_md` accessor: its internal `cat()` call is suppressed and its invisible return value is reused directly, so the combined report is emitted exactly once regardless of the number of species or components involved. As documented for the underlying accessors, a component with no data for a given species contributes nothing to that species' block: there is no "no data" placeholder or empty heading, since each `_md` function returns `""` in that case. `pacha_sc_full_name_md()` has no such empty case: it always renders a heading, falling back to the requested name verbatim when resolution fails.

A structurally invalid element of species (missing, or lacking at least a genus and a specific epithet) raises an error via the same validation used by the underlying accessors; species preceding the invalid one in the vector are not processed once that error is raised.

Value

A character scalar with the combined Markdown for every element of species, one block per species separated by a blank line. Returned invisibly when `print = TRUE`, visibly when `print = FALSE`.

Examples

```
## Not run:
# Inside a knitr chunk with `results = "asis"`:
pacha_report("Bidens andicola")
pacha_report(c("Bidens andicola", "Bomarea multiflora"), language = "en")
pacha_report("Bidens andicola", source = "coldp", use = "medicinal")

## End(Not run)
```

<code>pacha_sc_full_name</code>	<i>Full scientific name of a species</i>
---------------------------------	--

Description

Resolves species against ChecklistBank's Catalogue of Life dataset ("3LXR") via `match/nameusage`, falling back to `nameusage/search` (`limit = 100`) when no direct match is returned. Among search candidates the best match is chosen by preference: an exact case-insensitive name match with status "accepted"; else any exact name match; else the first "accepted" candidate; else the first candidate returned. `pacha_sc_full_name` returns the name as plain text; `pacha_sc_full_name_md` renders it as a Markdown heading (see "Functions"). This resolution always uses dataset "3LXR" and ignores `pacha_configure()`.

Usage

```
pacha_sc_full_name(species)
```

```
pacha_sc_full_name_md(species)
```

Arguments

<code>species</code>	Character scalar: a validated binomial name (non-NA, non-empty, at least two whitespace-separated tokens). Internal whitespace is collapsed and the string trimmed before use.
----------------------	--

Details

Results are cached in memory per lower-cased, whitespace-normalised species string, shared by both functions, so repeat lookups skip the HTTP call.

Failures never raise an error: network/HTTP errors, timeouts, malformed JSON, no match, or a missing **httr2** dependency all fall back to species verbatim with no authorship – the same output produced for a name that resolved but has no authorship of its own. `stop()` is only reached for a structurally invalid species.

Value

Character scalar: "Genus species Authorship" if authorship resolved, otherwise "Genus species" (or the original species if resolution failed). See "Functions" for the Markdown variant.

Functions

- `pacha_sc_full_name_md()`: Same resolution, rendered as a Markdown heading with the name italicised and the authorship appended as plain text (e.g. "## **Bidens andicola** Kunth"). Markdown-sensitive characters in both parts are backslash-escaped to keep the output well-formed. Printed via `cat()` and returned invisibly.

Examples

```
## Not run:
pacha_sc_full_name("Bidens andicola")
pacha_sc_full_name_md("Bidens andicola")

## End(Not run)
```

reference_pacha

*Retrieve the source database citation***Description**

Retrieves how the configured data source itself should be cited, as distinct from any per-species bibliographic reference. For the API source, this is the `citation` field of the configured ChecklistBank dataset (GET /dataset/{key}), resolved once and cached, since it does not vary by species. For the local source, this is the `citation` entry of `metadata.yaml` inside the configured ColDP archive. Either source may return the citation as CSL HTML (e.g. wrapped in `<div class="csl-entry">`, with `<span style="font-style: italic">` and HTML entities); this function strips all markup and decodes entities, returning clean, unformatted text.

Usage

```
reference_pacha(
  species = NULL,
  language = NULL,
  source = NULL,
  refresh = FALSE
)
```

Arguments

<code>species</code>	Ignored. Accepted only so a positional call does not collide with <code>language</code> . Unlike the other accessors in this module, <code>reference_pacha()</code> does not return a species-specific citation: the citation identifies the configured dataset or archive as a whole, and is identical for every species queried against that source. Passing a non-NULL value here has no effect on the result and emits a warning.
<code>language</code>	Optional output-language code or alias used for the localized no-data/connection message.
<code>source</code>	Optional source override: <code>"api"/"web"</code> or <code>"coldp"/"local"</code> . NULL uses the configured default.
<code>refresh</code>	Logical scalar. If TRUE, fetches the citation again instead of reusing its cached value.

Value

A single character string with the plain-text source citation, or the localized no-data/connection message.

Examples

```
## Not run:
reference_pacha()
reference_pacha(source = "coldp")

## End(Not run)
```

sustainable_uses_pacha

Retrieve sustainable-use records

Description

Retrieves sustainable-use categories and values from the configured source: the ChecklistBank taxon-property endpoint for the API source, or TaxonProperty.tsv for the local source. In either case, categories matching use_exclude_pattern are dropped. This keeps the function usable with any compatible property-based dataset, while remaining well suited to the Ecuadorian sustainable-use dataset used in the examples below.

Usage

```
sustainable_uses_pacha(
  species,
  use = NULL,
  language = NULL,
  source = NULL,
  refresh = FALSE
)

sustainable_uses_pacha_md(
  species,
  use = NULL,
  language = NULL,
  source = NULL,
  refresh = FALSE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
use	Optional category name or fragment, matched case-insensitively. NULL returns every available category.
language	Optional output-language code or alias used for labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.

Value

sustainable_uses_pacha() prints and invisibly returns a Category: heading followed by one indented use per line, for every matching category, or the localized no-data/connection message. sustainable_uses_pacha_md() returns, invisibly, the emitted Markdown string, or "" when no matching sustainable-use data are available.

See Also

[accessors_pacha](#) for a detailed description of plain-text and Markdown accessors.

Examples

```
## Not run:
sustainable_uses_pacha("Bidens andicola")
sustainable_uses_pacha("Bidens andicola", use = "medicinal", source = "coldp")

## End(Not run)
## Not run:
sustainable_uses_pacha_md("Bidens andicola", use = "medicinal")

## End(Not run)
```

threat_status_pacha	<i>Retrieve conservation threat status</i>
---------------------	--

Description

Retrieves the conservation-status value documented for a taxon. The API source reads the ChecklistBank taxon-distribution endpoint; the local source reads Distribution.tsv using the same country-aware, deterministic selection rule as [establishment_pacha\(\)](#). Typical values are IUCN categories such as EX, EW, CR, EN, VU, NT, LC, DD, and NE, though the function accepts any status recorded in the underlying dataset.

Usage

```
threat_status_pacha(species, language = NULL, source = NULL, refresh = FALSE)

threat_status_pacha_md(
  species,
  language = NULL,
  source = NULL,
  refresh = FALSE
)
```

Arguments

species	Character scalar containing at least a genus and specific epithet.
language	Optional output-language code or alias used to translate the threat category and labels.
source	Optional source override: "api"/"web" or "coldp"/"local". NULL uses the configured default ("api").
refresh	Logical scalar. If TRUE, fetches the selected source again instead of reusing its cached record.

Details

Values are translated through the selected language dictionary's threat mapping. Unmapped source values are returned in sentence case rather than rejected, so alternative assessment schemes remain usable.

Value

`threat_status_pacha()` prints and invisibly returns a translated conservation-status category, or the localized no-data/connection message. `threat_status_pacha_md()` returns, invisibly, the emitted Markdown string, or "" when the source has no threat-status record.

See Also

[accessors-pacha](#) for a detailed description of plain-text and Markdown accessors.

Examples

```
## Not run:
threat_status_pacha("Bidens andicola")
threat_status_pacha("Bidens andicola", source = "coldp", language = "en")

## End(Not run)
## Not run:
threat_status_pacha_md("Bidens andicola", language = "en")

## End(Not run)
```

Index

accessors-pacha, [2](#), [3](#), [6](#), [7](#), [16](#), [17](#)

common_names_pacha, [3](#)
common_names_pacha_md
 (common_names_pacha), [3](#)
common_names_pacha_md(), [11](#), [12](#)
compare_pacha, [4](#)

establishment_pacha, [5](#)
establishment_pacha(), [16](#)
establishment_pacha_md
 (establishment_pacha), [5](#)

indexation_urls_pacha, [6](#)
indexation_urls_pacha_md
 (indexation_urls_pacha), [6](#)
is_listed_pacha, [7](#)

pacha_clear_cache, [8](#)
pacha_configure, [9](#)
pacha_report, [11](#)
pacha_sc_full_name, [12](#)
pacha_sc_full_name_md
 (pacha_sc_full_name), [12](#)
pacha_sc_full_name_md(), [11](#), [12](#)

reference_pacha, [14](#)

sustainable_uses_pacha, [15](#)
sustainable_uses_pacha_md
 (sustainable_uses_pacha), [15](#)
sustainable_uses_pacha_md(), [11](#), [12](#)

threat_status_pacha, [16](#)
threat_status_pacha_md
 (threat_status_pacha), [16](#)