Package 'whoapi'

September 13, 2016

Type Package

Title A 'Whoapi' API Client

Version 0.1.2

Date 2016-09-12

Author Oliver Keyes

Maintainer Oliver Keyes <ironholds@gmail.com>

Description Retrieve data from the 'Whoapi' (https://whoapi.com) store of
domain information, including a domain's geographic location, registration
status and search prominence.

License MIT + file LICENSE

LazyData TRUE

Imports httr

Suggests testthat, knitr

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-09-13 11:45:39

R topics documented:

---

certificate_info   *Get Certificate Information for a Domain*

Description

certificate_info retrieves information about a domain's SSL certificate (if present).

Usage

```
certificate_info(token, domain, ...)
```

Arguments

token	a token generated with [whoapi_token](#)
domain	a domain name
...	further arguments to pass to httr's GET.

See Also

[whois_info](#) for information about the domain in general.

Examples

```
token <- whoapi_token("demokey")
## Not run:
whoapi_cert <- certificate_info(token, "whoapi.com")

## End(Not run)
```

---

domain_location   *Retrive Domain Location Information*

Description

domain_location returns geographic information about where a domain - or, specifically, its IP address - is located.

Usage

```
domain_location(token, domain, ...)
```

Arguments

token	a token generated with [whoapi_token](#)
domain	a domain name
...	further arguments to pass to httr's GET.

See Also

[whois_info](#) for more free-form information, including (potentially) the address of the domain holders.

Examples

```
token <- whoapi_token("demokey")
## Not run:
location_data <- domain_location(token, "whoapi.com")

## End(Not run)
```

---

domain_metadata   *Retrieve Domain Metadata*

Description

`domain_metadata` retrieves information about the content on a domain, specifically the title and metadata description fields from its home page

Usage

```
domain_metadata(token, domain, ...)
```

Arguments

token	a token generated with [whoapi_token](#)
domain	a domain name
...	further arguments to pass to httr's GET.

Examples

```
token <- whoapi_token("demokey")
## Not run:
metadata <- domain_metadata(token, "whoapi.com")

## End(Not run)
```

---

domain_rank   *Identify a Domain's Search Ranking*

Description

domain_rank provides users with the ability to identify a domain's "search ranking" - how promi- nent it is according to common internet benchmarks. Specifically, it provides both Alexa reach and popularity scores, and the Google Pagerank ("pr") ranking, which goes from 0 to 10 (and is represented by -1 if it cannot be found).

Usage

```
domain_rank(token, domain, ...)
```

Arguments

token	a token generated with whoapi_token
domain	a domain name
...	further arguments to pass to httr's GET.

See Also

domain_search which specifically looks at the search results a particular domain pulls up in various search engines.

Examples

```
token <- whoapi_token("demokey")
## Not run:
whoapi_domain <- domain_rank(token, "whoapi.com")

## End(Not run)
```

---

domain_search   *Identify a Domain's Search Results Count*

Description

domain_search allows you to quickly calculate, in an automated way, how many search results Bing and Google return for a particular domain.

Usage

```
domain_search(token, domain, ...)
```

Arguments

`token`	a token generated with [whoapi_token](#)
`domain`	a domain name
`...`	further arguments to pass to httr's GET.

See Also

[domain_rank](#) which looks more generally at domain popularity according to Alexa and Google Pagerank scores.

Examples

```
token <- whoapi_token("demokey")
## Not run:
search_results <- domain_search(token, "whoapi.com")

## End(Not run)
```

`is_blacklisted`	*Identify if a Domain is Blacklisted*

Description

`is_blacklisted` checks whether a domain is on prominent spam blacklists (or not).

Usage

```
is_blacklisted(token, domain, ...)
```

Arguments

`token`	a token generated with [whoapi_token](#)
`domain`	a domain name
`...`	further arguments to pass to httr's GET.

Value

a list containing a boolean value, "blacklisted", which identifies whether a domain is blacklisted by *any* of the checked services, followed by a breakdown of what the status is on each particular service.

---

is_taken   *Identify if a Domain is Taken*

Description

`is_taken` identifies if a domain is taken - if it's been reserve or registered.

Usage

```
is_taken(token, domain, ...)
```

Arguments

token	a token generated with [whoapi_token](#)
domain	a domain name
...	further arguments to pass to httr's GET.

Value

a boolean TRUE or FALSE indicating, respectively, whether the domain is reserved/registered, or not.

See Also

[whois_info](#) for more information about a domain's status, including when the registration expires and who has registered it.

Examples

```
#Check if whoapi.com is taken
token <- whoapi_token("demokey")
## Not run:
is_taken(token, "whoapi.com")

## End(Not run)
#[1] TRUE
```

---

whoapi   *Client library for 'Whoapi' (https://whoapi.com/)*

Description

y

---

whoapi_token   *Create a token for WhoAPI*

Description

This function allows the creation of connectors to whoapi, which can be passed between function calls, providing consistent API keys and options for different calls.

Usage

```
whoapi_token(key, user_agent = NULL)
```

Arguments

key	the API key to use.
user_agent	the user agent to use. NULL by default, which will use the whoapi package user agent.

---

whois_info   *Get Registration Information for a Domain*

Description

whois_info grabs detailed information about a domain's registration, including (but not limited to) who it's registered to, what its status is, when it was registered and when it expires.

Usage

```
whois_info(token, domain, ...)
```

Arguments

token	a token generated with whoapi_token
domain	a domain name
...	further arguments to pass to httr's GET.

See Also

is_taken for simpy determining if a domain is registered at all, or certificate_info for information about a domain's SSL certificates.

Examples

```
token <- whoapi_token("demokey")
## Not run:
whoapi_whois_info <- whois_info(token, "whoapi.com")

## End(Not run)
```

Index