

Package ‘wordpress’

April 23, 2021

Type Package

Title An API Wrapper for WordPress Site APIs

Version 0.2.2

URL <https://sixjupiter.com/wordpress/>

BugReports <https://github.com/simitpatel/wordpressr/issues>

Description An API wrapper that helps R developers retrieve data from a WordPress site's API in tidy format, in addition to publishing content on the WordPress site (WordPress API documentation: <<https://developer.wordpress.org/rest-api/>>).

Imports httr, tibble, dplyr, tidyr, magrittr, glue, purrr

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.0

Suggests knitr, rmarkdown, testthat, stringi

VignetteBuilder knitr

NeedsCompilation no

Author Simit Patel [aut, cre],
SixJupiter [cph, fnd]

Maintainer Simit Patel <simit@sixjupiter.com>

Repository CRAN

Date/Publication 2021-04-23 15:00:03 UTC

R topics documented:

create_wp_page	2
create_wp_post	3
create_wp_tags	5
delete_wp_page	6
delete_wp_tag	7
get_wp_authors	7

get_wp_cats	8
get_wp_pages	9
get_wp_posts	9
get_wp_posts_after	10
get_wp_tags	11
get_wp_users	11
update_wp_cat	12
update_wp_page	13
update_wp_post	14

Index	16
--------------	-----------

create_wp_page	<i>Create a New WordPress Page</i>
----------------	------------------------------------

Description

Creates a new page on the WordPress site provided using the credentials entered, with content and other information provided.

Usage

```
create_wp_page(
    root_url,
    user,
    pass,
    title_val,
    excerpt_val = "",
    content_val,
    status_val,
    slug_val = NULL,
    author_val,
    format_val = "standard"
)
```

Arguments

root_url	The domain on which you wish to create the page.
user	The username to be passed into the API call to create the page.
pass	The password to be used in the API call to create the page. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
title_val	The title of the page you are creating.
excerpt_val	The excerpt to be shown where your WordPress features excerpts of pages.
content_val	The content of the page.

status_val	The status of the page. Can be one of 'draft','publish','pending','future','private'.
slug_val	The slug to be assigned to the page. Can be automatically generated if left as NULL.
author_val	The user ID of the author creating the page.
format_val	The WordPress format to use. Defaults to 'standard'.

Value

A tibble containing the arguments passed in as well as the URL retrieved from the API response.

Examples

```
## Not run:
create_wp_page(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
title_val = 'post title',excerpt_val = 'post excerpt',
content_val = 'the post content as a string, with wordpress-accepted
<strong>html</strong> (or bbcode!)',
status_val = 'draft',format_val = 'standard')

## End(Not run)
```

create_wp_post	<i>Create a New WordPress Page</i>
----------------	------------------------------------

Description

wCreates a new post on the WordPress site provided using the credentials entered, with content and other information provided.

Usage

```
create_wp_post(
  root_url,
  user,
  pass,
  title_val,
  excerpt_val = "",
  content_val,
  fifu_val = NULL,
  status_val,
  author_val,
  format_val = "standard",
  categories_val,
  tag_val = ""
)
```

Arguments

root_url	The domain on which you wish to create the post.
user	The username to be passed into the API call to create the post.
pass	The password to be used in the API call to create the post. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
title_val	The title of the page you are creating.
excerpt_val	The excerpt to be shown where your WordPress features excerpts of post.
content_val	The content of the post.
fifu_val	If the Featured Image From URL plugin is installed, users can specify a remotely hosted image file to use as the featured image for the post. This field defaults to a value of NULL.
status_val	The status of the post. Can be one of 'draft', 'publish', 'pending', 'future', 'private'.
author_val	The user ID of the author creating the post.
format_val	The WordPress format to use. Defaults to 'standard'.
categories_val	The category IDs the post is to be associated with; comma separate in a character string if more than one.
tag_val	The tag IDs the post is to be associated with; comma separate in a category string if more than one.

Value

A tibble containing the arguments passed in as well as the URL retrieved from the API response.

Examples

```
## Not run:
create_wp_post(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
title_val = 'post title',excerpt_val = 'post excerpt',
content_val = 'the post content as a string, with wordpress-accepted
<strong>html</strong> (or bbcode!)',
fifu_val = 'https://domain.com/image.png',
status_val = 'draft',format_val = 'standard',categories_val = 1, tag_val = 1)

## End(Not run)
```

create_wp_tags	<i>Create a New WordPress Tag</i>
----------------	-----------------------------------

Description

Create a tag on your WordPress site using your WordPress site's API.

Usage

```
create_wp_tags(root_url, user, pass, tag_name, description_text)
```

Arguments

root_url	The domain on which you wish to create the post.
user	The username to be passed into the API call to create the post.
pass	The password to be used in the API call to create the post. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
tag_name	The text string associated with the tag.
description_text	The description of the tag; this value will be used wherever the description of the WordPress tag is used (perhaps in your theme).

Value

response from the API. 200 means the tag was created!

Examples

```
## Not run:
create_wp_tag(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
tag_name = 'cool posts',description_text = 'this is the description text for the tag "cool
posts". ')

## End(Not run)
```

delete_wp_page	<i>Delete a WordPress Page</i>
----------------	--------------------------------

Description

Delete a page on your WordPress site using your WordPress site's API.

Usage

```
delete_wp_page(root_url, user, pass, page_id, verbose = TRUE)
```

Arguments

root_url	The domain on which you wish to delete the page.
user	The username to be passed into the API call to delete the page.
pass	The password to be used in the API call to delete the page. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
page_id	The ID in the WordPress database of the page to be deleted. You can obtain IDs by using the <code>get_wp_pages()</code> function.
verbose	If TRUE, the page id will be printed to the console. Potentially useful for usage in a single-threaded loop.

Value

response from the API as a list object. Look for the key "deleted" within the list to confirm the post was deleted.

Examples

```
## Not run:  
delete_wp_tag(root_url = 'https://domain.com', user = Sys.getenv('username'),  
pass = Sys.getenv('password'), page_id = 12)  
  
## End(Not run)
```

delete_wp_tag	<i>Delete a WordPress Tag</i>
---------------	-------------------------------

Description

Delete a tag on your WordPress site using your WordPress site's API.

Usage

```
delete_wp_tag(root_url, user, pass, tag_id, verbose = TRUE)
```

Arguments

root_url	The domain on which you wish to create the post.
user	The username to be passed into the API call to create the post.
pass	The password to be used in the API call to create the post. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
tag_id	The text string associated with the tag.
verbose	If TRUE, the tag id will be printed to the console. Potentially useful for usage in a single-threaded loop.

Value

response from the API. 200 means the tag was deleted!

Examples

```
## Not run:  
delete_wp_tag(root_url = 'https://domain.com', user = Sys.getenv('username'),  
pass = Sys.getenv('password'), tag_id = 12)  
  
## End(Not run)
```

get_wp_authors	<i>Get WordPress Authors</i>
----------------	------------------------------

Description

Retrieves a list of all publishers on the site.

Usage

```
get_wp_authors(root_url)
```

Arguments

root_url The WordPress site for which the data is sought.

Value

A dataframe with four values: the user ID for the author in the site's WordPress database; the author's URL on the site; the author's name in the database; and the value of the description field associated with the author.

Examples

```
## Not run:  
get_wp_authors('domain.com')  
  
## End(Not run)
```

get_wp_cats *Get WordPress Categories*

Description

Retrieves the categories created on a WordPress site.

Usage

```
get_wp_cats(root_url)
```

Arguments

root_url The domain for which categories are sought to be retrieved.

Value

A data frame, unique by category ID, with 7 columns: category ID, category name, category slug, post count, description, URL, and parent category.

Examples

```
## Not run:  
get_wp_cats(root_url = 'domain.com')  
  
## End(Not run)
```

get_wp_pages	<i>Retrieve WordPress Pages</i>
--------------	---------------------------------

Description

Retrieve pages made on the WordPress site.

Usage

```
get_wp_pages(root_url, page_count = Inf, after_date = NULL)
```

Arguments

root_url	The WordPress site for which pages are sought to be retrieved.
page_count	The maximum number of pages to return, sorted by the most recent date. Default value is to return all pages made on the site.
after_date	The date after which pages should be returned.

Value

A data frame returning the post ID, publication date, title, excerpt, content, tag IDs, category IDs, and author IDs.

Examples

```
## Not run:  
get_wp_pages(root_url = 'https://domain.com', page_count = 200, after_date = NULL)  
  
## End(Not run)
```

get_wp_posts	<i>Retrieve WordPress Posts</i>
--------------	---------------------------------

Description

Retrieve posts made on the WordPress site.

Usage

```
get_wp_posts(root_url, post_count = Inf, after_date = NULL)
```

Arguments

root_url	The WordPress site for which posts are sought to be retrieved.
post_count	The maximum number of posts to return, sorted by the most recent date. Default value is to return all posts made on the site.
after_date	The date after which posts should be returned.

Value

A data frame returning the post ID, publication date, title, excerpt, content, tag IDs, category IDs, and author IDs.

Examples

```
## Not run:
get_wp_posts(root_url = 'https://domain.com', post_count = 200, after_date = NULL)

## End(Not run)
```

get_wp_posts_after *Retrieve WordPress Posts After a Certain Date*

Description

Retrieve posts made on the WordPress site.

Usage

```
get_wp_posts_after(root_url, after_date)
```

Arguments

root_url	The WordPress site for which posts are sought to be retrieved.
after_date	The date after which posts should be returned. Should be in "YYYY-MM-DD" format

Value

A data frame returning the post ID, publication date, title, excerpt, content, tag IDs, category IDs, and author IDs.

Examples

```
## Not run:
get_wp_posts(root_url = 'https://domain.com', post_count = 200, after_date = NULL)

## End(Not run)
```

`get_wp_tags`*Get WordPress Tags*

Description

Retrieves a list of all tags created on a given WordPress site.

Usage

```
get_wp_tags(root_url)
```

Arguments

`root_url` The WordPress site for which the data is sought.

Value

A dataframe with three values: the slug for a given tag, its ID in the WordPress site database, and the human readable tag name.

Examples

```
## Not run:  
get_wp_tags('domain.com')  
  
## End(Not run)
```

`get_wp_users`*Retrieve WordPress User List*

Description

Returns all the registered users on a given WordPress site that can be retrieved without a key.

Usage

```
get_wp_users(root_url)
```

Arguments

`root_url` The domain for which users are sought to be retrieved.

Value

A dataframe with three columns: the ID of the user in the WordPress database; their user name on the site; and the URL of their user page on the site.

Examples

```
## Not run:
get_wp_users(root_url = 'domain.com')

## End(Not run)
```

update_wp_cat	<i>Update a WordPress Category</i>
---------------	------------------------------------

Description

Update the name, description, or slug of a category in your WordPress site.

Usage

```
update_wp_cat(
  domain,
  user,
  pass,
  tag_id,
  tag_name,
  tag_slug,
  tag_description,
  verbose = TRUE
)
```

Arguments

domain	The domain on which you wish to update the given category.
user	The username to be passed into the API call to update the category.
pass	The password to be used in the API call to update the category. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
tag_id	The ID of the category desired to be updated in the WordPress site's database.
tag_name	The desired name of the category that users will see on the WordPress site.
tag_slug	The slug to be associated with the WordPress category. Change with caution, as changing the slug may result in URLs being broken.
tag_description	The description of the category; this value will be used wherever the description of the WordPress category is used (perhaps in your theme).
verbose	Defaults to TRUE; determines whether or not the desired category name and slug will be printed in the console. Potentially useful if employing this function in a loop in which many categories will be updated at once.

Value

response from the API. 200 means success – the category was updated!

Examples

```
## Not run:
update_wp_cat(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
tag_name = 'cool posts',description_text = 'this is the description text for the category
"very cool posts". ')

## End(Not run)
```

update_wp_page	<i>Update WordPress Page</i>
----------------	------------------------------

Description

Updates an existing WordPress page with new values.

Usage

```
update_wp_page(
  root_url,
  user,
  pass,
  page_id,
  title_val,
  excerpt_val = "",
  fifu_val,
  content_val,
  status_val,
  author_val
)
```

Arguments

root_url	The domain on which you wish to create the page.
user	The username to be passed into the API call to create the page.
pass	The password to be used in the API call to create the page. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
page_id	The numeric ID in the WordPress database associated with the page being updated.

title_val	The title of the page being updated.
excerpt_val	The excerpt to be shown where your WordPress features excerpts of pages.
fifu_val	If the FIFU plugin is installed, the URL of the featured image can be inserted here.
content_val	The content of the page.
status_val	The status of the page. Can be one of 'draft','publish','pending','future', 'private'.
author_val	The user ID of the author to be associated with the page.

Value

A list containing the status code of the API call. A status code of 200 indicates the call was a success.

Examples

```
## Not run:
update_wp_page(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
page_id = 123,title_val = 'post title',excerpt_val = 'post excerpt',
fifu_val = 'https://remotesite.com/image.png',
content_val = 'the page content as a string, with wordpress-accepted
<strong>html</strong> (or bbcode!)',
status_val = 'draft',author_val = '2')

## End(Not run)
```

update_wp_post	<i>Update a WordPress Post</i>
----------------	--------------------------------

Description

Updates an existing WordPress post with new values.

Usage

```
update_wp_post(
  root_url,
  user,
  pass,
  title_val,
  excerpt_val = "",
  content_val,
  fifu_val = NULL,
  status_val,
  post_id,
```

```

    author_val,
    format_val = "standard",
    categories_val,
    tag_val = ""
)

```

Arguments

root_url	The domain on which you wish to create the post.
user	The username to be passed into the API call to create the post.
pass	The password to be used in the API call to create the post. To get this value, you must have the Application Passwords plugin installed, and must create an application using that plugin via your WordPress admin panel; there you will get the password needed.
title_val	The title of the page you are creating.
excerpt_val	The excerpt to be shown where your WordPress features excerpts of post.
content_val	The content of the post.
fifu_val	If the Featured Image From URL plugin is installed, users can specify a remotely hosted image file to use as the featured image for the post. This field defaults to a value of NULL.
status_val	The status of the post. Can be one of 'draft','publish','pending','future', 'private'.
post_id	The ID of the post to be updated.
author_val	The user ID of the author creating the post.
format_val	The WordPress format to use. Defaults to 'standard'.
categories_val	The category IDs the post is to be associated with; comma separate in a character string if more than one.
tag_val	The tag IDs the post is to be associated with; comma separate in a category string if more than one.

Value

A list containing the status code of the API call. A status code of 200 indicates the call was a success.

Examples

```

## Not run:
create_wp_post(root_url = 'https://domain.com',user = Sys.getenv('username'),
pass = Sys.getenv('password'),
title_val = 'post title',excerpt_val = 'post excerpt',
content_val = 'the post content as a string, with wordpress-accepted
<strong>html</strong> (or bbcode!)',
fifu_val = 'https://domain.com/image.png',
status_val = 'draft',post_id = 123,format_val = 'standard',categories_val = 1, tag_val = 1)

## End(Not run)

```

Index

[create_wp_page](#), [2](#)
[create_wp_post](#), [3](#)
[create_wp_tags](#), [5](#)

[delete_wp_page](#), [6](#)
[delete_wp_tag](#), [7](#)

[get_wp_authors](#), [7](#)
[get_wp_cats](#), [8](#)
[get_wp_pages](#), [9](#)
[get_wp_posts](#), [9](#)
[get_wp_posts_after](#), [10](#)
[get_wp_tags](#), [11](#)
[get_wp_users](#), [11](#)

[update_wp_cat](#), [12](#)
[update_wp_page](#), [13](#)
[update_wp_post](#), [14](#)