

# Package ‘boxr’

August 16, 2016

**Type** Package

**Title** Interface for the 'Box.com API'

**Version** 0.3.3

**URL** <https://github.com/brendan-r/boxr/>

**BugReports** <https://github.com/brendan-r/boxr/issues>

**Description** An R interface for the remote file hosting service 'Box' (<<https://www.box.com/>>). In addition to uploading and downloading files, this package includes functions which mirror base R operations for local files, (e.g. `box_load()`, `box_save()`, `box_read()`, `box_setwd()`, etc.), as well as 'git' style functions for entire directories (e.g. `box_fetch()`, `box_push()`).

**License** MIT + file LICENSE

**Imports** htr (>= 1.1.0), httpuv, assertthat, stringr, dplyr, digest, bit64, rio, mime, utils, stats

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 5.0.1

**NeedsCompilation** no

**Author** Brendan Rocks [aut, cre]

**Maintainer** Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

**Repository** CRAN

**Date/Publication** 2016-08-16 09:57:15

## R topics documented:

<code>boxr-package</code>	2
<code>boxr_options</code>	3
<code>boxr_S3_classes</code>	3
<code>box_add_description</code>	4
<code>box_auth</code>	5

box_auth_on_attach . . . . .	6
box_delete_file . . . . .	7
box_dir_create . . . . .	8
box_dl . . . . .	9
box_fetch . . . . .	10
box_fresh_auth . . . . .	12
box_ls . . . . .	12
box_previous_versions . . . . .	13
box_read . . . . .	14
box_save . . . . .	15
box_search . . . . .	16
box_setwd . . . . .	17
box_source . . . . .	18
box_write . . . . .	19
<b>Index</b>	<b>20</b>

---

boxr-package	<i>boxr</i>
--------------	-------------

---

## Description

A light-weight, high-level R interface for the box.com API

## Details

An introduction to the package, and a short getting started guide (including how to authorize boxr to work with your box.com account), can be found in the package's vignette. Simply run

```
vignette("boxr")
```

and it should open in your browser.

To see a list of the functions which boxr provides, click on the "Index" hyperlink at the bottom of this page.

boxr's home on the internet is currently GitHub: <https://github.com/brendan-r/boxr>. If you find anything you think might be a bug, do report it as a [GitHub issue!](#)

Happy hacking! :)

## Author(s)

Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

---

boxr_options	<i>Display global options for boxr</i>
--------------	--

---

### Description

Run `boxr_options()` to see what it's possible to set globally.

### Usage

```
boxr_options()
```

### Details

Options can be set in the usual way using [options](#), and include:

`box.verbose` Should boxr print to the console using `cat`? This is slightly 'rude' package behaviour, and may cause problems if using the `knitr` package.

`box.wd` A list containing the name and id of the default box.com directory

`box.token` The token object used for authentication

### Value

A list of the options available

### Author(s)

Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

### See Also

[box\\_setwd](#) for another way to set the default box.com directory

---

boxr_S3_classes	<i>boxr S3 Classes</i>
-----------------	------------------------

---

### Description

boxr has a few very simple S3 classes for the data returned by the API. While `httr` returns objects in its own system of classes, generally boxr extracts information, and converts the JSON response to an R list with `httr::content` (perhaps with some information judged to be extraneous removed). If you'd rather get to the list itself, you can do this with `unclass(x)`.

The following classes are used:

**boxr\_file\_reference** Returned by `box_ul`, and similar functions (e.g. `box_save`). A description of a file remotely hosted on box.com. Available methods: `print`.

**boxr\_folder\_reference** As above, but for folders/directories. Available methods: print

**boxr\_object\_list** Returned by [box\\_search](#), and related functions. A list, with each entry being a reference to a file or folder hosted on box.com . Available methods: print, for a summary of the first few results, and as .data.frame, to coerce some of the API response's information to a [data.frame](#).

**boxr\_dir\_comparison** Returned by the internal function [box\\_dir\\_diff](#). Available methods: print, summary.

**boxr\_dir\_wide\_operation\_result** Returned by [box\\_fetch](#) and [box\\_push](#). Available methods: print, summary

### Author(s)

Brendan Rocks <foss@brendanrocks.com>

---

box\_add\_description    *Add a description to a file hosted remotely on box.com*

---

### Description

Files hosted on box.com can have small descriptions underneath their filenames. This can be useful to explain the contents of the file, or even to leave 'git commit' style messages about the latest changes made to them.

### Usage

```
box_add_description(file_id, description)
```

### Arguments

`file_id`            The box.com id for the file that you'd like to download  
`description`        The description which you'd like to add to the file. [character](#).

### Value

An object of class [boxr\\_file\\_reference](#).

### Author(s)

Brendan Rocks <foss@brendanrocks.com>

---

box_auth	<i>Authenticate a box.com Account</i>
----------	---------------------------------------

---

## Description

box\_auth serves two purposes:

1. Setting up box.com accounts with boxr for the first time
2. Connecting to previously used box.com accounts

In either case, it should be sufficient to run `box_auth()` with no parameters. If you've authenticated with boxr before, this should be all that is required. However, the first time you use boxr, the process is slightly more involved (see 'Getting Set-Up' below).

## Usage

```
box_auth(client_id = "", client_secret = "", interactive = TRUE,
         cache = "~/boxr-oauth", write.Renv = TRUE, ...)
```

## Arguments

<code>client_id</code>	Optional. The client id for the account you'd like to use. character.
<code>client_secret</code>	Optional. The client secret for the account you'd like to use. character.
<code>interactive</code>	logical. Should the authorization process happen interactively (requiring user input to the R console, and/or a visit to box.com)?
<code>cache</code>	Passed to cache in <a href="#">httr</a> .
<code>write.Renv</code>	logical. If authentication was successful, should <code>client_id</code> and <code>client_secret</code> be written to <code>.Renvirons</code> in your HOME directory?
<code>...</code>	Passed to <a href="#">oauth2.0_token</a>

## Value

Returns TRUE if connection successful, throws an error otherwise. Invoked for its side effect; OAuth2.0 connection to the box.com API.

Your `client_id` and `client_secret` will be written to `~/Renviron` for future use, and a token object will be written to the path supplied by `cache` (`~/boxr-oauth` by default).

See [oauth2.0\\_token](#) for details.

## Getting Set-Up

A version of this guide is in the package vignette, with some additional screenshots. To view the vignette, run `vignette("boxr")`.

To use boxr for the first time, you need to enable API access for your box.com account. The process is slightly annoying. You only need to do it once - it takes around 2 minutes.

The next time you use boxr, you should be able to just run `box_auth()` (without entering anything else) to be authenticated and ready-to-go.

At <https://www.box.com/developers>, click on 'My Apps', in the top right hand corner log in and create a new 'app' for your box.com account. This won't create an app of any description; you'll simply be granting yourself programmatic access to your files. You can call it anything you like.

1. **'Create an app'** **Set OAuth2 Parameters** On the next screen, you'll want to check the box for **Content API Access Only**, and enter `http://localhost` as your **redirect\_uri**.
3. **Connect boxr to your account** Run `box_auth()` and pass your `client_id` and `client_secret` to the console when prompted. These strings are not enough for someone to access your account maliciously. However, it's still a good idea to keep them safe, and out of any files or code which might be shared with others.

A browser window should open, for you to formally grant yourself access to your files at box.com. From this point on, simply running `box_auth()` at the start of a session should be all that is required.

### Author(s)

Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

### See Also

[box\\_auth\\_on\\_attach](#), [oauth2.0\\_token](#) for details of how the tokens are handled

---

box_auth_on_attach	<i>Authenticate box.com account automatically</i>
--------------------	---

---

### Description

This function saves you the effort of typing `box_auth()` after the package loads. Executing `box_auth_on_attach(TRUE)` will mean that boxr will automatically attempt to authorize itself when 'attached' (e.g. `library(boxr)`), using the credentials from the current session.

### Usage

```
box_auth_on_attach(auth_on_attach = FALSE)
```

### Arguments

`auth_on_attach` Should boxr try and connect to your account when attached? logical

### Value

Nothing; invoked for it's side effect.

**Note**

This is provided for convenience, but it's a bad idea to use, if:

**You'd like your code to be reproducible** Even if your collaborators have access to the same files on box.com, as the default behaviour is to require using `box_auth()`, code is likely to become irreproducible.

**You use more than one box.com account** Things could get rather confusing.

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_auth](#)

---

box_delete_file	<i>Move files to and from the box.com trash folder</i>
-----------------	--

---

**Description**

- `box_delete_file` moves a file to the trash folder.
- `box_delete_folder` moves a whole folder (and all of it's contents) to the trash folder.
- `box_restore_file` moves a file from the trash, to wherever it was before
- `box_delete_folder` does the same thing for a folder.

**Usage**

```
box_delete_file(file_id)
```

```
box_delete_folder(dir_id)
```

```
box_restore_folder(dir_id)
```

```
box_restore_file(file_id)
```

**Arguments**

<code>file_id</code>	The box.com id for the file that you'd like delete/restore
<code>dir_id</code>	The box.com id for the folder that you'd like delete/restore

**Details**

'Deleting' a file in this case means moving it to a special folder within your box.com account called the 'Trash'. At the time of writing, the files are stored for three months before being deleted, and the contents of the folder can be accessed from the top right hand menu of the web interface.

**Value**

An object of class [boxr\\_file\\_reference](#).

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_ul](#), for putting files there in the first place

---

box_dir_create	<i>Create a new box.com folder</i>
----------------	------------------------------------

---

**Description**

Create a new box.com folder

**Usage**

```
box_dir_create(dir_name, parent_dir_id = box_getwd())
```

**Arguments**

`dir_name`            The name for the directory you'd like to create.  
`parent_dir_id`        The box.com folder id of the folder you'd like your new folder to be within.

**Value**

An object of class [boxr\\_folder\\_reference](#).

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_delete\\_folder](#) to delete remote folders/directories, [box\\_ls](#) to examine their contents.



---

 box\_dl

*Download and upload individual files from box.com*


---

## Description

Functions to download (`box_dl`), and upload (`box_ul`).

`box_dl` takes the id of a file hosted on box.com, downloads it and writes it to disk.

`box_ul` uploads a file stored locally to a specified box.com folder. If a file with the same name already exists, it will upload the file as a new version.

## Usage

```
box_dl(file_id, local_dir = getwd(), overwrite = FALSE, filename = NULL,
       version_id = NULL, version_no = NULL, pb = options()$boxr.progress)
```

```
box_ul(dir_id = box_getwd(), file, pb = options()$boxr.progress,
       description = NULL)
```

## Arguments

<code>file_id</code>	The box.com id for the file that you'd like to download
<code>local_dir</code>	A file path to a local directory which you'd like the file to be downloaded to.
<code>overwrite</code>	logical. Should existing files with the same name be overwritten?
<code>filename</code>	Optional. An alternate filename for the local version of the file. The default, NULL, uses the name from box.com.
<code>version_id</code>	If downloading an older version, the <code>version_id</code> of the desired file
<code>version_no</code>	The version of the file you'd like to download (starting at 1)
<code>pb</code>	Should a progress bar be shown? (via <a href="#">setTxtProgressBar</a> )
<code>dir_id</code>	If uploading, the box.com folder id that you'd like to upload to.
<code>file</code>	the path to the local file that you'd like to upload (if there is one)
<code>description</code>	Optional. character. A string to be used as the description caption for the file (added via <a href="#">box_add_description</a> ). Useful for describing the contents of a file, or describing the latest changes made to it. If NULL (the default), no description is added.

## Value

`box_dl` returns the path of the newly downloaded file if successful, and throw an error otherwise.

`box_ul` will return an object of class [boxr\\_file\\_reference](#)

**Versions**

box\_dl can accept one of two parameters to specify file versions: `version_id` and `version_no`.

The box.com API refers to file versions using 11 digit ids (which can be accessed via [box\\_previous\\_versions](#)) - you can specify these using the `version_id` parameter.

However, this isn't terribly intuitive. As a result, box\_dl provides the `version_no` parameter, which accepts a whole number, and corresponds to the versions that you'll see via the web UI. For example to download the version marked 'V2' on box.com, specify `version_no = 2`. This works by making an internal call to [box\\_previous\\_versions](#) to retrieve the `version_id`, which makes it slightly slower.

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_fetch](#) and [box\\_push](#) for directory-wide equivalents, [box\\_delete\\_file](#) for removing uploaded files, [box\\_source](#) for R code, and [box\\_save/box\\_load](#) for remote R objects.

---

box\_fetch

*Recursive, Directory-wide Operations to Synchronize Local and box.com directories*

---

**Description**

These functions take a path to a local directory, and a box.com folder id, and perform synchronization operations.

box\_fetch downloads the contents of a box.com folder to a local directory.

box\_push uploads the contents of local directory to a box.com folder.

Files which are present in the origin but not the destination will be copied over.

Behaviour when a file exists in both depends on the parameters described below.

**Usage**

```
box_fetch(dir_id = box_getwd(), local_dir = getwd(), recursive = TRUE,
  overwrite = FALSE, delete = FALSE)
```

```
box_push(dir_id = box_getwd(), local_dir = getwd(), ignore_dots = TRUE,
  overwrite = FALSE, delete = FALSE)
```

**Arguments**

dir_id	The id for the box.com folder
local_dir	The path to the local directory
recursive	logical. Should the call include subdirectories and thier contents?
overwrite	Where the same files exist in both the origin and the destination, and the files in the origin are newer, should the files in the destination be updated (overwritten)?
delete	logical. Should files which exist in the destination, but not the origin, be deleted?
ignore_dots	logical. Should local directories with filenames begining with dots be ignored? This is useful for 'invisible' folders such as .git and .Rproj.user where up-loading them may be unexpected.

**Details****Overwrite/Update**

In the interests of preventing mishaps, `overwrite` is by default set to `FALSE`, which means that files which exist in the destination, but which are out of date, are not modified.

Setting `overwrite` to `TRUE` is likely to produce expected behavior for most users.

This is a conservative precaution to prevent users unexpectedly overwriting their files, and may change as a default in later releases.

However, files which are updated on box.com are versioned, and most operating systems have file recovery features (e.g. 'Trash' (Ubuntu/Debian/OSX), or 'Recycle Bin' (Windows)), so unintended modification of files will be revertable for most users.

**Implementation**

At the time of writing, the box.com API only allows for one file at a time to be uploaded/downloaded, and as a result, `boxr` recursively scans the directory tree, uploading/downloading files in loops. Because the box.com API can send, but not accept, gzipped files, downloading tends to be faster than uploading.

`box_fetch/box_push` rely on the internal function `box_dir_diff` to determine how to process individual files (e.g. which to update, which to leave as is, etc.). See it's help page for details.

**Value**

An object of class `boxr_dir_wide_operation_result`, describing the file operations performed

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

`box_dl/box_ul` for single file operations. `box_dir_diff` is the internal function which determines how files should be processed.

---

box_fresh_auth	<i>Reauthorise a Problematic box.com connection</i>
----------------	---

---

**Description**

Very simply, deletes the old token file before trying to re-authorise. This is often the solution to authorisation problems raised by users!

**Usage**

```
box_fresh_auth(cache = "~/boxr-oauth", ...)
```

**Arguments**

cache	Passed to cache in <a href="#">httr</a> .
...	Passed to <a href="#">box_auth</a>

**See Also**

[box\\_auth](#) for the usual method of authorisation, and [box\\_auth\\_on\\_attach](#) for a lazy one.

---

box_ls	<i>Obtain a data.frame describing the contents of a box.com folder</i>
--------	--

---

**Description**

Obtain a data.frame describing the contents of a box.com folder

**Usage**

```
box_ls(dir_id = box_getwd())
```

**Arguments**

dir_id	The box.com id for the folder that you'd like to query
--------	--

**Value**

A data.frame describing the contents of the the folder specified by dir\_id. Non recursive.

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_fetch](#) and [box\\_push](#) for synchronizing the contents of local and remote directories. [list.files](#) for examining the contents of local directories.

---

box\_previous\_versions *Retrieve details of previous versions of a remote file*

---

## Description

box.com explicitly versions files. `box_previous_versions` returns a `data.frame` containing information on a file's previous versions on box.com. No information about the current version of the file is returned.

## Usage

```
box_previous_versions(file_id)
```

## Arguments

`file_id`            the id of the file you'd like version information about

## Value

A `data.frame` containing information about previous versions of the file (if available). Importantly, it contains the `file_version_id`, which can be passed to `box_dl`.

## Author(s)

Brendan Rocks <foss@brendanrocks.com>

## References

This function is a light wrapper for box.com API's `versions` method.

<https://developers.box.com/docs/#files-view-versions-of-a-file>

## See Also

[box\\_dl](#)

---

box_read	<i>Read files from box.com into memory as R objects</i>
----------	---

---

### Description

box\_read will download a file specified by file\_id, and attempt to read it into memory as an R object. This can be useful, for example, to read in a .csv file as a `data.frame`.

Converting a file to an R object is by default handled by `rio`'s `import` function. The only modification of its behavior is that json files are not necessarily coerced to `data.frames`, but can be used to store list data, too. In addition, more specific read functions are provided:

box\_read\_csv Reads remote .csv files as `data.frames` (via `read.csv`)

box\_read\_tsv Reads remote .tsv files as `data.frames` (via `read.delim`)

box\_read\_json Reads remote .json files as `lists` (via `toJSON`)

box\_read\_excel Reads remote Microsoft Excel files as `data.frames` (via `read_excel`)

### Usage

```
box_read(file_id, type = NULL, version_id = NULL, version_no = NULL,
         read_fun = rio::import, fread = FALSE, ...)
```

```
box_read_csv(file_id, ...)
```

```
box_read_tsv(file_id, ...)
```

```
box_read_json(file_id, ...)
```

```
box_read_excel(file_id, ...)
```

### Arguments

file_id	The box.com id for the file that you'd like to download
type	MIME type (aka internet media type) used to override the content type returned by the server. See <a href="http://en.wikipedia.org/wiki/Internet_media_type">http://en.wikipedia.org/wiki/Internet_media_type</a> for a list of common types
version_id	If downloading an older version, the version_id of the desired file
version_no	The version of the file you'd like to download (starting at 1)
read_fun	The function used to read the data into R. Defaults to <code>rio::import</code>
fread	Should the function <code>data.table::fread</code> be used to read .csv files? Passed to <code>rio::import</code> (if used).
...	Passed to as additional parameters to read_fun

### Author(s)

Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

**See Also**

[box\\_dl](#) for saving files to disk, [box\\_save](#) for working with R workspaces, and [box\\_source](#) for working with R code.

---

box_save	<i>Save and load R workspaces via box.com</i>
----------	---

---

**Description**

These convenience functions aim to provide analogous functionality to [load](#) and [save.image](#) (or [save](#)), but for .RData files stored on box.com, as opposed to locally.

**Usage**

```
box_save(..., dir_id = box_getwd(), file_name = ".RData",
         description = NULL)
```

```
box_save_image(dir_id = box_getwd(), file_name = ".RData",
              description = NULL)
```

```
box_load(file_id)
```

**Arguments**

...	The objects to be saved. Quoted or unquoted. Passed to <a href="#">save</a> .
dir_id	The box.com folder id where the objects will be stored as a .RData file.
file_name	The name you'd like your .Rdata file saved as. For example, "myworkspace.RData"
description	Optional. character. A string to be used as the description caption for the file (added via <a href="#">box_add_description</a> ). Useful for describing the contents of a file, or describing the latest changes made to it. If NULL (the default), no description is added.
file_id	For <a href="#">box_load</a> , the box.com id of the .RData or .rda file you'd like to load into your workspace.

**Details**

[box\\_save](#) saves an .RData file using [save.image](#) if objects is not supplied or [save](#) if it is. The file is then uploaded to box.com via [box\\_ul](#).

[box\\_load](#) downloads a file from box.com using [box\\_dl](#), and then [loads](#) it into the current workspace.

**Value**

[box\\_load](#) returns a character vector of the names of objects created, invisibly. [box\\_save](#) and [box\\_save\\_image](#) are used for their side effects, and don't return anything.

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

The base R functions which these wrap; [save](#), [save.image](#) and [load](#).

---

box\_search

*Search the files in a box.com account*

---

**Description**

Search the files in a box.com account

**Usage**

```
box_search(query = "", content_types = c("name", "description",
    "file_content", "comments", "tags"), type = NULL, file_extensions = NULL,
    ancestor_folder_ids = NULL, created_at_range = NULL,
    updated_at_range = NULL, size_range = NULL, trash = FALSE,
    owner_user_ids = NULL, auto_paginate = TRUE, max = 200)
```

```
box_search_files(query, ...)
```

```
box_search_folders(query, ...)
```

```
box_search_trash(query, ...)
```

**Arguments**

query	The search term that you'd like to use
content_types	The content types you'd like to search. More than one can be supplied with a vector
type	The type of object you'd like returned. The default of NULL return all possible types. Other acceptable values are "file", "folder", or "weblink"
file_extensions	A vector of strings containing the file extensions (without dots) by which to narrow your search.
ancestor_folder_ids	If supplied, results are limited to one or more parent folders
created_at_range	Optional. Limit search to a range of created at times. A vector of two dates (coercible via <a href="#">as.POSIXct</a> ).
updated_at_range	Optional. Limit search to a range of updated at times. A vector of two dates (coercible via <a href="#">as.POSIXct</a> ).



size_range	Optional. Limit search to a range of file sizes, in bytes. A vector of two whole numbers (coerible via (coercible via <a href="#">as.numeric</a> and <a href="#">round</a> )).
trash	Should the search be limited to the trash folder? <a href="#">logical</a> .
owner_user_ids	Optional. Limit search to a files owned by a set of users. A vector if IDs, coercible with <a href="#">as.integer64</a> .
auto_paginate	<a href="#">logical</a> . By default, the box.com will return a fixed number of search results per request. By setting auto_paginate to TRUE, boxr will keep making new requests untill all search results are in.
max	<a href="#">numeric</a> . Upper limit on the number of search results returned (protective measure for users with large numbers of files).
...	Parameters passed to box_search

### Details

The box.com API supports a maximum of 200 results per request. If `max > 200`, then multiple requests will be sent to retrieve and combine 'paginated' results for you, behind the scenes.

See the [box.com search description](#) for details of the features of the service. Some notable details:

**Full Text Searching** Is available for many source code file types, though not including R at the time of writing.

**Boolean Operators Are Supported** Such as and, or, and not (upper or lower case).

**Phrases Can Be Searched** By putting them in "quotation marks".

**Search Availability** It takes around 10 minutes for a newly uploaded file to enter the search index.

### Value

An object of class `boxr_object_list`. See [boxr\\_S3\\_classes](#) for details.

### Author(s)

Brendan Rocks <[foss@brendanrocks.com](mailto:foss@brendanrocks.com)>

---

box\_setwd

*Get/Set Default box.com directory/folder*

---

### Description

Providing analgous functionality for the jbase R functions [getwd](#) and [setwd](#), these functions set and retrieve a default box.com `dir_id`, stored in [boxr\\_options](#).

### Usage

```
box_setwd(dir_id)
```

```
box_getwd()
```

**Arguments**

dir\_id            The id of the folder you'd like to set as your default

**Value**

box\_getwd returns the id of the default folder. box\_setwd does nothing and is used for its side-effects.

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_ls](#) to examine the contents of a remote directory, and [box\\_fetch/box\\_push](#) for synchronizing them.

---

box\_source

*Execute R code stored on box.com*

---

**Description**

box\_source will download a remote R script stored on box.com, and then attempt to execute it, using [source](#).

**Usage**

```
box_source(file_id)
```

**Arguments**

file\_id            The box.com id for the file that you'd like to download

**Author(s)**

Brendan Rocks <foss@brendanrocks.com>

**See Also**

[box\\_dl](#) for saving files to disk, [box\\_save](#) for working with R workspaces, and [box\\_read](#) for reading files into memory as R objects.

---

`box_write`*Write R Objects to Files Remotely Hosted on box.com*

---

### Description

A fast and lazy way to upload R objects to box.com in a commonly readable file format. `read_fun` is used to convert R objects to files, which by default is the `export` function from the `rio` package. `rio`'s `export` function currently only supports `data.frames`; for lists `toJSON` may be more appropriate.

Note: `box_write` is for writing files in standard formats to box.com. To upload R objects as `.RData` files, see `box_save`.

### Usage

```
box_write(x, filename, dir_id = box_getwd(), description = NULL,
          write_fun = rio::export, ...)
```

### Arguments

<code>x</code>	An R object
<code>filename</code>	The name for the file to be uploaded
<code>dir_id</code>	If uploading, the box.com folder id that you'd like to upload to.
<code>description</code>	Optional. character. A string to be used as the description caption for the file (added via <code>box_add_description</code> ). Useful for describing the contents of a file, or describing the latest changes made to it. If <code>NULL</code> (the default), no description is added.
<code>write_fun</code>	The function used to write the R object to a file
<code>...</code>	Additional arguments passed to <code>read_fun</code>

### Value

An object of class `boxr_file_reference`.

### Author(s)

Brendan Rocks <foss@brendanrocks.com>

# Index

## \*Topic **package**

boxr-package, 2

as.integer64, 17

as.numeric, 17

as.POSIXct, 16

box\_add\_description, 4, 9, 15, 19

box\_auth, 5, 6, 7, 12

box\_auth\_on\_attach, 6, 6, 12

box\_delete\_file, 7, 10

box\_delete\_folder, 8

box\_delete\_folder (box\_delete\_file), 7

box\_dir\_create, 8

box\_dir\_diff, 4, 11

box\_dl, 9, 11, 13, 15, 18

box\_fetch, 4, 10, 10, 12, 18

box\_fresh\_auth, 12

box\_getwd (box\_setwd), 17

box\_load, 10

box\_load (box\_save), 15

box\_ls, 8, 12, 18

box\_previous\_versions, 10, 13

box\_push, 4, 10, 12, 18

box\_push (box\_fetch), 10

box\_read, 14, 18

box\_read\_csv (box\_read), 14

box\_read\_excel (box\_read), 14

box\_read\_json (box\_read), 14

box\_read\_tsv (box\_read), 14

box\_restore\_file (box\_delete\_file), 7

box\_restore\_folder (box\_delete\_file), 7

box\_save, 3, 10, 15, 15, 18, 19

box\_save\_image (box\_save), 15

box\_search, 4, 16

box\_search\_files (box\_search), 16

box\_search\_folders (box\_search), 16

box\_search\_trash (box\_search), 16

box\_setwd, 3, 17

box\_source, 10, 15, 18

box\_ul, 3, 8, 11, 15

box\_ul (box\_dl), 9

box\_write, 19

boxr (boxr-package), 2

boxr-package, 2

boxr\_file\_reference, 4, 8, 9, 19

boxr\_folder\_reference, 8

boxr\_options, 3, 17

boxr\_S3\_classes, 3, 17

cat, 3

character, 4

content, 3

data.frame, 4, 13, 14

export, 19

getwd, 17

httr, 3, 5, 12

import, 14

list, 14

list.files, 12

load, 15, 16

logical, 17

oauth2.0\_token, 5, 6

options, 3

read.csv, 14

read.delim, 14

read\_excel, 14

rio, 14, 19

round, 17

save, 15, 16

save.image, 15, 16

setTxtProgressBar, 9

setwd, [17](#)

source, [18](#)

toJSON, [14](#), [19](#)