
LuaGit2 Documentation

Release 1.0.0

Satyendra Kumar Banjare

Jun 25, 2019

1	Compatibility	3
1.1	Installation	3
1.2	Basic Documentation	4
1.3	Examples	105

Luagit2 is a tool to do cool git things on lua. It depends on **Libgit2 Version 0.27.0** and consists of updated Lua bindings to the same.

It is currently compatible to be used only for `lua 5.3`.

It is built using **Lua_C_Api** and has its tests written using **Busted**. It can be easily installed using luarocks and Cmake/Make.

- Libgit2 - **0.27.0**
- Lua- **5.3**
- **Operating systems :**
 1. Linux
 2. MacOS

1.1 Installation

This section is about how to successfully install `luagit2` from the source.

1.1.1 Requirements :

- Lua 5.3 (ONLY)
- Libgit2 - 0.27.0
- `liblua5.3-dev`

First make sure `Libgit2` is successfully installed on your system. There are some important libraries on which `Libgit2` depends.

Here is a quick handy guide to install them

```
$ sudo apt-get install libhttp-parser-dev
$ sudo apt install libssh2-1-dev
$ sudo apt install libcurl4-openssl-dev
```

Finally install `Libgit2` using :

```
$ wget https://github.com/libgit2/libgit2/archive/v0.27.0.tar.gz
$ tar xzf v0.27.0.tar.gz
$ cd libgit2-0.27.0/
$ cmake .
$ make
$ sudo make install
```

Install liblua5.3-dev using :

```
$ sudo apt-get install liblua5.3-dev
```

At this point you should be all ready for installation of luagit2.

Do a quick git clone of luagit2 using :

```
$ git clone https://github.com/SatyendraBanjare/luagit2.git
```

The user may now proceed with installation using any of these two methods.

1.1.2 Cmake/Make

In the root directory of luagit2.

```
$ mkdir build && cd build
$ cmake ..
$ make
$ sudo make install
```

1.1.3 Luarocks

In the root directory of luagit2.

```
$ sudo luarocks make luagit2-0.0-1.rockspec
```

If you also have busted installed on your system, you may proceed to tests.

1.2 Basic Documentation

This section contains explanation on Luagit2 API.

1.2.1 Annotated

These are the luagit2 methods that help to produce annotation commit for the tracked files in a given git repository.

These annotation commits can then be used while doing merge, rebase or creating a branch from an annotated commit. Refer to this https://libgit2.org/libgit2/#v0.27.0/type/git_annotated_commit.

Contents

- *annotated_commit_from_ref*

- *annotated_commit_from_revspec*
- *annotated_commit_lookup*
- *annotated_commit_from_fetchhead*
- *annotated_commit_id*
- *annotated_commit_free*

annotated_commit_from_ref

parameters	Returns
luagit2_repository	luagit2_annotated_commit
reference_name(string)	

1. `luagit2_repository`: The repository to look into for the reference
2. `reference_name`: the `reference_name` string value using which the annotated commit will be created.
 - Creates a `luagit2_annotated_commit` from given reference which should be free using *annotated_commit_free*.
 - *libgit2 equivalent*: https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_from_ref

annotated_commit_from_revspec

parameters	Returns
luagit2_repository	luagit2_annotated_commit
luagit2_oid	

1. `luagit2_repository`: The repository to look into for the given oid.
2. `luagit2_oid`: the oid to look for and to be used to create annotated commit.
 - Creates a `luagit2_annotated_commit` from given revspec which should be free using *annotated_commit_free*.
 - *libgit2 equivalent*: https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_from_revspec

annotated_commit_lookup

parameters	Returns
luagit2_repository	luagit2_annotated_commit
luagit2_oid	

1. `luagit2_repository`: The repository to look into for the annotated commit's oid.
2. `luagit2_oid`: the annotated commit's oid to lookup for.
 - Creates a `luagit2_annotated_commit` from given annotated commit's oid which should be free using *annotated_commit_free*.
 - *libgit2 equivalent*: https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_lookup

annotated_commit_from_fetchhead

parameters	Returns
luagit2_repository	luagit2_annotated_commit
branch_name (string)	
remote_url_name (string)	
luagit2_oid	

1. `luagit2_repository` : The repository to look into for provided values.
 2. `branch_name` : The branch name of whose remote url is to be used.
 3. `remote_url_name` : Name of remote url name.
 4. `luagit2_oid` : the commit oid (preferably latest) of provided remote branch.
- Creates a `luagit2_annotated_commit` from given fetchhead data which should be free using *annotated_commit_free*.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_from_fetchhead

annotated_commit_id

parameters	Returns
luagit2_annotated_commit	luagit2_oid

1. `luagit2_annotated_commit` : The annotated commit whose oid is to be found.
- Creates a `luagit2_oid` from given `annotated_commit`.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_id

annotated_commit_free

parameters	Returns
luagit2_annotated_commit	None

1. `luagit2_annotated_commit` : The annotated commit which is to be freed.
- Frees a `luagit2_annotated_commit`.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/annotated/git_annotated_commit_free

1.2.2 Blame

These methods help get the blame data for the files.

Get the blame data as the last commit that introduced that change of line in a file.

Contents

- *blame_file*
- *blame_get_hunk_byindex*

- *blame_get_hunk_byline*
- *blame_get_hunk_count*
- *blame_free*

blame_file

Get the blame data for a file.

Parameters	Returns
luagit2_repository	luagit2_blame
File_path (string)	

1. `luagit2_repository`: The repository to look into for the file.
2. `File_path`: the file path whose blame data is to be found.
 - Creates `luagit2_blame` or gives an error.

blame_get_hunk_byindex

Parameters	Returns
luagit2_blame	luagit2_blame_hunk
index_number (integer)	

1. `luagit2_blame`: the blame data using which blame hunk will be calculated.
2. `index_number`: The index number of entry to be used.
 - Returns corresponding `luagit2_blame_hunk` or gives an error.

Note: After Getting Blame_Hunk , One may use helper methods **blame_hunk_commit_id** & **blame_hunk_signature** to find the last commit's oid and committer.

blame_get_hunk_byline

Parameters	Returns
luagit2_blame	luagit2_blame_hunk
line_number (integer)	

1. `luagit2_blame`: the blame data using which blame hunk will be calculated.
2. `line_number`: The line number in the file for which blame is to be found.
 - Returns corresponding `luagit2_blame_hunk` or gives an error.

Note: After Getting Blame_Hunk , One may use helper methods **blame_hunk_commit_id** & **blame_hunk_signature** to find the last commit's oid and committer.

blame_get_hunk_count

parameters	Returns
luagit2_blame	count_num (integer)

1. `luagit2_blame` : The blame data for which blame hunk is to be found.
 - Returns corresponding count of blame hunks in that blame data.

blame_free

parameters	Returns
luagit2_blame	None

1. `luagit2_blame` : The blame data which is to be freed.

1.2.3 Blob

Contents

- *blob_create_fromdisk*
- *blob_create_fromworkdir*
- *blob_filtered_content*
- *blob_id*
- *blob_is_binary*
- *blob_lookup*
- *blob_lookup_prefix*
- *blob_owner*
- *blob_rawsize*
- *blob_free*

blob_create_fromdisk

parameters	Returns
luagit2_repository	luagit2_oid
path_to_blob(string)	

1. `luagit2_repository` : The repository to look into for the given blob.
2. `path_to_blob` : the path to the blob in repository .

Note: The path should be relative to current path, which may or may not inside the directory.

- Creates a `luagit2_oid` from blob's path.

- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_create_fromdisk

blob_create_fromworkdir

parameters	Returns
luagit2_repository	luagit2_oid
path_to_blob(string)	

1. `luagit2_repository` : The repository to look into for the given blob.
2. `path_to_blob` : the path to the blob in repository.

Note: The path should be relative to root directory of repository. Prefer this method over **blob_create_fromdisk** while creating blob using its path and not oid.

- Creates a `luagit2_oid` from blob's path.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_create_fromdisk

blob_filtered_content

parameters	Returns
luagit2_blob	luagit2_buf
path_to_blob(string)	
check_for_binary_data(int)	

1. `luagit2_blob` : The `luagit2_blob` whose content to look for .
2. `path_to_blob` : the path to the blob in repository.
3. `check_for_binary_data` : integer value of bool whether to check or not for filtered content.

Note: The path should be relative to root directory of repository.

- Creates a `luagit2_buf` from blob's Content, can be read using **buf_details()**.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_filtered_content

blob_id

parameters	Returns
luagit2_blob	luagit2_oid

1. `luagit2_blob` : The blob whose oid is to be found.
- Creates a `luagit2_oid` for given blob.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_id

blob_is_binary

parameters	Returns
luagit2_blob	boolean

1. `luagit2_blob` : The blob which is to be checked.
 - returns boolean if blob is binary or not.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_is_binary

blob_lookup

parameters	Returns
luagit2_repository	luagit2_blob
luagit2_oid	

1. `luagit2_repository` : The repository to look into for the given blob.
2. `luagit2_oid` : the oid of the blob to look for.
 - returns `luagit2_blob` for given oid or an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_lookup

blob_lookup_prefix

parameters	Returns
luagit2_repository	luagit2_blob
luagit2_oid	
length (int)	

1. `luagit2_repository` : The repository to look into for the given blob.
2. `luagit2_oid` : the oid of the blob to look for.
3. `length` : length of oid to use for looking up.
 - returns `luagit2_blob` for given oid or an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_lookup_prefix

blob_owner

parameters	Returns
luagit2_blob	luagit2_repository

1. `luagit2_blob` : The blob whose owner repo is to be found.
 - returns `luagit2_repository` or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_owner

blob_rawsize

parameters	Returns
luagit2_blob	size(number)

1. `luagit2_blob` : The blob whose content size is to be found.

- returns a numerical value of the blob's size or gives an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_rawsize

blob_free

parameters	Returns
luagit2_blob	None

1. `luagit2_blob` : The blob which is to be freed.

- returns None or gives an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/blob/git_blob_free

1.2.4 Branch

These methods allows user to work with `git` branches of a given repository.

These help to create, delete, iterate over list of branches and setting upstreams for a branch.

Contents

- *branch_create*
- *branch_create_from_annotated*
- *branch_delete*
- *branch_is_checked_out*
- *branch_is_head*
- *branch_iterator_new*
- *branch_lookup*
- *branch_move*
- *branch_name*
- *branch_next*
- *branch_set_upstream*
- *branch_upstream*

branch_create

parameters	Returns
luagit2_repository	luagit2_reference
branch_name (string)	
luagit2_commit	
force (int)	

1. `luagit2_repository` : The repository to create a new branch in.
2. `branch_name` : the name of new branch to create .
3. `luagit2_commit` : the last commit using which the branch will be created.
4. `force` : integer equivalent of boolean whether or not force create a new branch.
 - Creates a `luagit2_reference` for the new created branch.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_create

branch_create_from_annotated

parameters	Returns
luagit2_repository	luagit2_reference
branch_name (string)	
luagit2_annotated_commit	
force (int)	

1. `luagit2_repository` : The repository to create a new branch in.
2. `branch_name` : the name of new branch to create .
3. `luagit2_annotated_commit` : the annotated commit using which the branch will be created.
4. `force` : integer equivalent of boolean whether or not force create a new branch.
 - Creates a `luagit2_reference` for the new created branch.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_create_from_annotated

branch_delete

parameters	Returns
luagit2_reference	None

1. `luagit2_reference` : The reference for the branch which is to be deleted.
 - returns none or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_delete

branch_is_checked_out

parameters	Returns
luagit2_reference	Boolean

1. `luagit2_reference` : The reference for the branch which to check.
 - returns boolean for whether branch is currently checked out or not.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_is_checked_out

branch_is_head

parameters	Returns
<code>luagit2_reference</code>	Boolean

1. `luagit2_reference` : The reference for the branch which to check.
 - returns boolean for whether branch is being pointed by HEAD or not.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_is_head

branch_iterator_new

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_branch_iterator</code>
<code>luagit2_branch_type</code>	

1. `luagit2_repository` : The repository to look for the list of branches.
2. `luagit2_branch_type` : the type of branches to look .It can be either LOCAL or REMOTE. It can be created using luagit2's branch helper methods `get_type_GIT_BRANCH_LOCAL` & `get_type_GIT_BRANCH_REMOTE` .
 - Creates a `luagit2_branch_iterator` for list of branches. The list index starts with 0.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_iterator_new

branch_lookup

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_reference</code>
<code>branch_name (string)</code>	
<code>luagit2_branch_type</code>	

1. `luagit2_repository` : The repository to look into for the given branch.
2. `branch_name` : the path to the blob in repository .
3. `luagit2_branch_type` : the type of branches to look . It can be either LOCAL or REMOTE. It can be created using luagit2's branch helper methods `get_type_GIT_BRANCH_LOCAL` & `get_type_GIT_BRANCH_REMOTE` .
 - Creates a `luagit2_reference` from looked up branch or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_lookup

branch_move

parameters	Returns
luagit2_reference	luagit2_reference
new_branch_name (string)	
force (int)	

1. `luagit2_reference` : The reference for the branch which is to be moved to a new name.
2. `new_branch_name` : The new branch name.
3. `force` : integer equivalent of bool to force branch moving.
 - Creates a `luagit2_reference` for new branch or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_move

branch_name

parameters	Returns
luagit2_reference	String (name)

1. `luagit2_reference` : The reference for the branch whose name is to be found.
 - returns name of branch or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_name

branch_next

parameters	Returns
luagit2_branch_type	luagit2_reference
luagit2_branch_iterator	

1. `luagit2_branch_type` : The type of branches to look .It can be either LOCAL or REMOTE. It can be created using luagit2's branch helper methods `get_type_GIT_BRANCH_LOCAL` & `get_type_GIT_BRANCH_REMOTE` .
2. `luagit2_branch_iterator` : The branch iterator that contains list of all branches .
 - Creates a `luagit2_reference` for next branch.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_next

branch_set_upstream

parameters	Returns
luagit2_reference	None
upstream_name(string)	

1. `luagit2_reference` : The reference for the branch.
2. `upstream_name` : The upstream's name.
 - returns none or an error.

- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_set_upstream

branch_upstream

parameters	Returns
luagit2_reference	luagit2_reference

1. `luagit2_reference` : The reference for the branch whose upstream branch is to be found.

- Creates a `luagit2_reference` of upstream branch or gives an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/branch/git_branch_upstream

1.2.5 Buf

Contents

- *buf_set_str*
- *buf_free*

buf_set_str

parameters	Returns
data (string)	luagit2_buf

1. `data` : The string data to put in the `luagit2_buf`.

- returns `luagit2_buf` or an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/buf/git_buf_set

buf_free

Note: The `buf_free` is being worked upon. It is advised not to use that for now.

parameters	Returns
data (string)	none

1. `luagit2_buf` : The `luagit2_buf` to free.

- returns `none` or an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/buf/git_buf_free

1.2.6 Checkout

These methods help doing a `git checkout` to either a repository's **HEAD** or *Index* or *Tree*.

Contents

- *checkout_init_options*
- *checkout_head*
- *checkout_index*
- *checkout_tree*

checkout_init_options

parameters	Returns
None	luagit2_checkout_options

- Creates a `luagit2_checkout_options` to be used in other functions or gives an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/checkout/git_checkout_init_options

checkout_head

parameters	Returns
luagit2_repository	None
luagit2_checkout_options	

1. `luagit2_repository` : The repository in which checkout is to be performed.
2. `luagit2_checkout_options` : The checkout options to be used while checkout is performed.
 - returns none or an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/checkout/git_checkout_head

checkout_index

parameters	Returns
luagit2_repository	None
luagit2_index	
luagit2_checkout_options	

1. `luagit2_repository` : The repository in which checkout is to be performed.
2. `luagit2_index` : The index to which checkout is to be performed.
3. `luagit2_checkout_options` : The checkout options to be used while checkout is performed.
 - returns none or an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/checkout/git_checkout_index

checkout_tree

parameters	Returns
luagit2_repository	None
luagit2_tree	
luagit2_checkout_options	

1. `luagit2_repository` : The repository in which checkout is to be performed.
2. `luagit2_tree` : The tree to which checkout is to be performed.
3. `luagit2_checkout_options` : The checkout options to be used while checkout is performed.

- returns none or an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/checkout/git_checkout_tree

1.2.7 Clone

These methods help in cloning a valid git repository.

clone_public

Note: Only **Public** repositories can be cloned using this function.

parameters	Returns
path_where_to_save(string)	None
url_to_clone (string)	
print_progress (int)	

1. `path_where_to_save` : The path where to clone the repository. The *git* folder will be saved at provided path. example: If you want to clone a repo at **example_folder** then pass this value as **example_folder/.git** so this example_folder will be top level directory for the cloned repository.
2. `url_to_clone` : The url value of git repo to clone
3. `print_progress` : The integer equivalent of whether or not to print progress.

- returns none or an error.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/clone/git_clone

1.2.8 Commit

These methods help in creating new git commits and extracting information about already created commits.

Contents

- *commit_author*
- *commit_body*
- *commit_committer*

- *commit_extract_signature*
- *commit_header_field*
- *commit_lookup*
- *commit_lookup_prefix*
- *commit_message*
- *commit_message_encoding*
- *commit_message_raw*
- *commit_nth_gen_ancestor*
- *commit_parent*
- *commit_parent_id*
- *commit_parentcount*
- *commit_raw_header*
- *commit_summary*
- *commit_time*
- *commit_tree*
- *commit_tree_id*
- *commit_free*
- *commit_id*
- *commit_create_update_head*
- *commit_create_update_none*
- *commit_create_with_signature*
- *commit_create_initial*

commit_author

parameters	Returns
luagit2_commit	luagit2_signature

1. `luagit2_commit` : The commit whose author is to be known.
 - Returns a `luagit2_signature` for commit's author or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_author

commit_body

parameters	Returns
luagit2_commit	body_data (string)

1. `luagit2_commit` : The commit whose contents are to be found out.
 - Returns a string for commit's content or gives an error.

- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_body

commit_committer

parameters	Returns
luagit2_commit	luagit2_signature

1. `luagit2_commit` : The commit whose committer is to be known.
- Returns a `luagit2_signature` for commit's committer or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_committer

commit_extract_signature

parameters	Returns
luagit2_repository	luagit2_buf (signature)
luagit2_oid	luagit2_buf (extra)

1. `luagit2_repository` : The repository to look in for the commit's details.
 2. `luagit2_oid` : The commit's oid to look for.
- Returns two `luagit2_buf` or gives an error. The first buf is for signature details and other is extra for the commit's details. Example :
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_extract_signature

Example:

```
local sign_buf, extra_buf = luagit2.commit_extract_signature(repo, commit_oid)
```

commit_header_field

parameters	Returns
luagit2_commit	luagit2_buf
field (string)	

1. `luagit2_commit` : The commit whose header field is to be known.
 2. `field` : the stringvalue of field to lookup for.
- returns `luagit2_buf` for the details or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_header_field

commit_lookup

parameters	Returns
luagit2_repository	luagit2_commit
luagit2_oid	

1. `luagit2_repository` : the repository in which commit is to be looked up.

2. `luagit2_oid`: the oid of commit to be looked up.
 - returns `luagit2_commit` for the commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_lookup

commit_lookup_prefix

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_commit</code>
<code>luagit2_oid</code>	
<code>length (int)</code>	

1. `luagit2_repository`: the repository in which commit is to be looked up.
2. `luagit2_oid`: the oid of commit to be looked up.
3. `length`: the number of first *n* characters to be used to look up for the commit in repository.
 - returns `luagit2_commit` for the commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_lookup_prefix

commit_message

parameters	Returns
<code>luagit2_commit</code>	<code>message_data (string)</code>

1. `luagit2_commit`: The commit whose message is to be found out.
 - Returns a string for commit's message or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_message

commit_message_encoding

parameters	Returns
<code>luagit2_commit</code>	<code>message_encoding (string)</code>

1. `luagit2_commit`: The commit whose message's encoding is to be found out.
 - Returns a string for commit's message encoding which is used or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_message_encoding

commit_message_raw

parameters	Returns
<code>luagit2_commit</code>	<code>raw_message_data (string)</code>

1. `luagit2_commit`: The commit whose message are to be found out.
 - Returns a string for commit's raw message or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_message_raw

commit_nth_gen_ancestor

parameters	Returns
luagit2_commit	luagit2_commit
nth_generation (int)	

1. `luagit2_commit` : The commit whose ancestors are to be found.
2. `nth_generation` : How old ancestor commit is to be found.
 - returns a `luagit2_commit` of the ancestor or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_nth_gen_ancestor

commit_parent

parameters	Returns
luagit2_commit	luagit2_commit
position (int)	

1. `luagit2_commit` : The commit whose parents are to be found.
2. `position` : How old parent should be. The *0th parent means immediate parent commit and '1* for one previous to immediate parent commit.
 - returns a `luagit2_commit` of the parent commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_parent

commit_parent_id

parameters	Returns
luagit2_commit	luagit2_oid
position (int)	

1. `luagit2_commit` : The commit whose parents are to be found.
2. `position` : How old parent should be. The *0th parent means immediate parent commit and '1* for one previous to immediate parent commit.
 - returns a `luagit2_oid` of the parent commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_parent_id

commit_parentcount

parameters	Returns
luagit2_commit	parent_count (int)

1. `luagit2_commit` : The commit whose number of parents are to be found.
 - returns the number of parent commits or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_parentcount

commit_raw_header

parameters	Returns
luagit2_commit	raw_header (string)

1. `luagit2_commit` : The commit whose raw header is to be found.
 - returns the raw header data for the commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_raw_header

commit_summary

parameters	Returns
luagit2_commit	commit_summary (string)

1. `luagit2_commit` : The commit whose message summary is to be found.
 - returns the message summary string for the commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_summary

commit_time

parameters	Returns
luagit2_commit	commit_time (string)

1. `luagit2_commit` : The commit whose commit time is to be found.
 - returns the `commit_time` string for the commit or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_time

commit_tree

parameters	Returns
luagit2_commit	luagit2_tree

1. `luagit2_commit` : The commit whose tree is to be found.
 - returns the `luagit2_tree` for the commit's parent tree or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_tree

commit_tree_id

parameters	Returns
luagit2_commit	luagit2_oid

1. `luagit2_commit` : The commit whose tree's oid is to be found.
 - returns the `luagit2_oid` for the commit's parent tree oid or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_tree_id

commit_free

parameters	Returns
luagit2_commit	None

1. `luagit2_commit` : The commit which is to be freed.
 - returns none or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_free

commit_id

parameters	Returns
luagit2_commit	luagit2_oid

1. `luagit2_commit` : The commit whose oid is to be found out.
 - returns `luagit2_oid` for the commit's oid or gives an error.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_id

commit_create_update_head

parameters	Returns
luagit2_repository	luagit2_oid
luagit2_signature(author)	
luagit2_signature(committer)	
message (string)	
luagit2_tree	
luagit2_commit (immediate parent commit)	

1. `luagit2_repository` : The repository where to create a new commit.
2. `luagit2_signature(author)` : The author's signature.
3. `luagit2_signature(committer)` : The committer's signature.
4. `Message` : The string message to be created with new commit.
5. `luagit2_tree` : The tree in which the commit is to be made.
6. `luagit2_commit` : The immediate parent commit which is to be used to create the new commit.

Note: The parent commit should not have any child commits. Also the parent commit should also belong to that same tree.

- returns a `luagit2_oid` of the newly created commit or gives an error. This method updates the **HEAD** tip of git repository so if you do a git log, it will show the output of latest commit.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_create_v

commit_create_update_none

parameters	Returns
luagit2_repository	luagit2_oid
luagit2_signature(author)	
luagit2_signature(committer)	
message (string)	
luagit2_tree	
luagit2_commit (immediate parent commit)	

1. `luagit2_repository` : The repository where to create a new commit.
2. `luagit2_signature(author)` : The author's signature.
3. `luagit2_signature(committer)` : The committer's signature.
4. `Message` : The string message to be created with new commit.
5. `luagit2_tree` : The tree in which the commit is to be made.
6. `luagit2_commit` : The immediate parent commit which is to be used to create the new commit.

Note: The parent commit should not have any child commits. Also the parent commit should also belong to that same tree.

- returns a `luagit2_oid` of the newly created commit or gives an error. This method **does not** update the **HEAD** tip of git repository so if you do a git log, it **will not** show the output of latest commit.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_create_v

commit_create_with_signature

parameters	Returns
luagit2_repository	luagit2_oid
commit_content(string)	
commit_sign(string)	

1. `luagit2_repository` : The repository where to create a new commit.
 2. `commit_content` : The content to be used to make a commit. It should contain details of parent, tree, author, committer and commit message.
 3. `commit_sign` : The committer's **pgp** signature.
- returns a `luagit2_oid` for the new created commit or gives an error. This method updates the **HEAD** tip of git repository so if you do a git log, it will show the output of latest commit.
 - *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_create_with_signature

Note: The data used for creating commit should be all valid. the parent commit should be immediate and of same tree.

Example :

```

local gpgsig = "-----BEGIN PGP SIGNATURE-----\n\
Version: GnuPG v1.4.12 (Darwin)\n\
\n\
iQIcBAABAgAGBQJQ+FMIAAoJEH+LfPdZDSs1e3EQAjMjhqjWF+WkGLHju7pTw2a1\n\
o6IoMAhv0Z/LHlWhzBd9e7JeCnanRt12bAU7yvYp9+Z+z+dbwqLwDoFp8LVuigl8\n\
JGLcnwiUW3rSvhjdCp9irdb4+bhKUnKUzSdsR2CK4/hC0N2i/HOvMYX+BRsvqweq\n\
AsAkA6dAWh+gAfedrBUkCTGhlNYoetjdakWqlGL1TiKAefEZrtAlTpPkGn92vbLq\n\
SphFRUY9hVn1ZBWrT3hEpvAicZag3rTOiRVT1X1flj8B2vGCEr3RrcwOIZikpdaW\n\
who/X3xh/DGbI2RbuxmmJpXXP/8dsVchRJzBwG+yhwU/iN3MlV2c5D69tIs/Dok\n\
6VbyU4lm/ae0y3yR83D9dUlkycOnmm1BAHKIZ9qUts9X7mWJf0+yy2QxJVpjaTGG\n\
cmnQKKPeNIhGJk2ENnnnzjEve7L7YJQF6itbx5VC0csGh30cb3YR7DMdWjt7f8pu\n\
c6j+q1rP7EpE2afUN/geSlp5i3x8aXZPDj67jImbVCE/Q1X9voCtyzGJH7MXR0N9\n\
ZpRF8yzveRfMH8bwAJjSOGAFF5XkcR/RNY95o+J+QcgBLdX48h+ZdNmUf6jqlu3J\n\
7KmTXXQcOVpN6dD3CmRFsbjq+x6RHwa8uliGn+oIkX908r97ckfB/kHKH7ZdXIJC\n\
cpxtDQQMGYFpXK/71stq\n\
=ozeK\n\
-----END PGP SIGNATURE-----";

local commit_data = "tree 1b49a09ae5e707b8ed48e6048aa1901138838944\n\
parent 108ddee361877aa5c044d89d8dd232b8fd0f8992\n\
author Test_user <test@example.com> 1358451456 -0800\n\
committer Test_user <test@example.com> 1358451456 -0800\n\
\n\
new commit which works\n";

local new_commit_id = luagit2.commit_create_with_signature(repo,data,gpgsig)

```

commit_create_initial

parameters	Returns
luagit2_repository	luagit2_oid
luagit2_signature(author)	
luagit2_signature(committer)	
message (string)	
luagit2_tree	

1. `luagit2_repository`: The repository where to create a new commit.
2. `luagit2_signature(author)`: The author's signature.
3. `luagit2_signature(committer)`: The committer's signature.
4. `Message`: The string message to be created with new commit.
5. `luagit2_tree`: The tree in which the initial commit is to be made.

Note: This is the first commit to be made in the repository. Take care that there is no previous commit present. The tree for this initial commit, pass value from `index_write_tree()`.

- returns a `luagit2_oid` of the newly created commit or gives an error. This method updates the **HEAD** tip of git repository so if you do a git log, it will show the output of latest commit.
- *libgit2 equivalent* : https://libgit2.org/libgit2/#v0.27.0/group/commit/git_commit_create_v

1.2.9 Config

These methods help to read and modify configuration details for a git repository.

Contents

- *config_delete_entry*
- *config_delete_multivar*
- *config_find_global*
- *config_find_programdata*
- *config_find_system*
- *config_find_xdg*
- *config_get_int32*
- *config_get_int64*
- *config_get_bool*
- *config_get_path*
- *config_get_string*
- *config_get_string_buf*
- *config_open_default*
- *config_open_global*
- *config_open_level*
- *config_open_ondisk*
- *config_parse_bool*
- *config_parse_int32*
- *config_parse_int64*
- *config_parse_path*
- *config_set_bool*
- *config_set_int32*
- *config_set_int64*
- *config_set_multivar*
- *config_set_string*
- *config_snapshot*
- *config_free*
- *config_add_file_ondisk*
- *config_new*
- *config_iterator_new*
- *config_entry_free*

- *config_get_entry*
- *config_next*
- *config_iterator_free*

config_delete_entry

parameters	Returns
luagit2_config	None
entry_name (string)	

1. `luagit2_config`: The config whose entry is to be deleted.
 2. `entry_name`: The entry name to delete.
- returns none or gives an error.

config_delete_multivar

parameters	Returns
luagit2_config	None
entry_name (string)	
regex_name (string)	

1. `luagit2_config`: The config whose entry is to be deleted.
 2. `entry_name`: The entry name to delete.
 3. `regex_name`: The regex using which to search.
- returns none or gives an error.

config_find_global

parameters	Returns
None	luagit2_buf

- There are no parameters required. It finds global git configuration details and returns `luagit2_buf` consisting of that data or gives an error.

config_find_programdata

parameters	Returns
None	luagit2_buf

- There are no parameters required. It finds programdata git configuration details and returns `luagit2_buf` consisting of that data or gives an error.

Note: This method should be used on windows system only.

config_find_system

parameters	Returns
None	luagit2_buf

- There are no parameters required. It finds system git configuration details and returns `luagit2_buf` consisting of that data or gives an error.

config_find_xdg

parameters	Returns
None	luagit2_buf

- There are no parameters required. It finds XDG git configuration details and returns `luagit2_buf` consisting of that data or gives an error.

Note: This method should be used on windows system only.

config_get_int32

parameters	Returns
luagit2_config	None
entry_name (string)	

1. `luagit2_config`: The config whose entries are to be looked up.
 2. `entry_name`: The entry whose value is to be found.
- Get the value of an 32bit integer config variable or en error.

config_get_int64

parameters	Returns
luagit2_config	None
entry_name (string)	

1. `luagit2_config`: The config whose entries are to be looked up.
 2. `entry_name`: The entry whose value is to be found.
- Get the value of an 64bit integer config variable or an error.

config_get_bool

parameters	Returns
luagit2_config	boolean
entry_name (string)	

1. `luagit2_config`: The config whose entries are to be looked up.

2. `entry_name` : The entry whose value is to be found.

- Get the boolean value of an config variable or an error.

config_get_path

parameters	Returns
<code>luagit2_config</code>	<code>luagit2_buf</code>
<code>entry_name</code> (string)	

1. `luagit2_config` : The config whose entries are to be looked up.

2. `entry_name` : The entry whose value is to be found.

- returns a `luagit2_buf` for path value of an config variable or an error.

config_get_string

parameters	Returns
<code>luagit2_config</code>	<code>luagit2_buf</code>
<code>entry_name</code> (string)	

1. `luagit2_config` : The config whose entries are to be looked up.

2. `entry_name` : The entry whose value is to be found.

- returns a string for value of an config variable or an error.

config_get_string_buf

parameters	Returns
<code>luagit2_config</code>	<code>luagit2_buf</code>
<code>entry_name</code> (string)	

1. `luagit2_config` : The config whose entries are to be looked up.

2. `entry_name` : The entry whose value is to be found.

- returns a `luagit2_buf` for string value of an config variable or an error.

config_open_default

parameters	Returns
None	<code>luagit2_config</code>

- There are no parameters required. It finds default git configuration details and returns `luagit2_config` consisting of that data or gives an error.

config_open_global

parameters	Returns
luagit2_config (parent)	luagit2_config

1. `luagit2_config`: The config whose global git config details are to be found.

- returns `luagit2_config` for given's config global values or an error.

config_open_level

parameters	Returns
luagit2_config (parent)	luagit2_config
luagit2_config_level_t	

1. `luagit2_config`: The parent config used to find values.

2. `luagit2_config_level_t`: The config level to open. This can be obtained by any of these helper methods `config_level_program_data()`, `config_level_system()`, `config_level_xdg()`, `config_level_global()` or `config_level_local()`.

Note: The xdg and programdata config level should be used on;y in windows type system.

- Returns corresponding `luagit2_config` or gives an error.

config_open_ondisk

parameters	Returns
path (string)	luagit2_config

1. `path`: The path of config file in the repository to look for. Onw may also open global config file by passing appropriate path value.

Note: One may also first create a config file using `config_add_file_ondisk()` and then use it.

- returns corresponding `luagit2_config` or an error.

config_parse_bool

parameters	Returns
value(string)	boolean

1. `value`: string value to parse.

- returns a boolen value for the passed string. The value true is returned for *true*, *yes*, *on* or *1* or any other number other than zero and false for *false*, *no*, *off* and *0*.

config_parse_int32

parameters	Returns
value(string)	number

1. value : string value to parse.
 - returns 32 bit int type number value of passed string.

config_parse_int64

parameters	Returns
value(string)	number

1. value : string value to parse.
 - returns 64 bit int type number value of passed string.

config_parse_path

parameters	Returns
value(string)	luagit2_buf

1. value : string value to parse.
 - returns luagit2_buf for passed path string value.

config_set_bool

Use this to set boolean values to entries in a git config.

parameters	Returns
luagit2_config	None
entry_name (string)	
value (boolean)	

1. luagit2_config : The config whose entries are to be set.
2. entry_name : The entry name to set.
3. value : boolean value to be set.
 - Returns none or gives an error.

config_set_int32

parameters	Returns
luagit2_config	None
entry_name (string)	
value (number)	

1. luagit2_config : The config whose entries are to be set.

2. `entry_name` : The entry name to set.
 3. `value` : 32 bit integer type value to be set.
- Returns none or gives an error.

config_set_int64

parameters	Returns
<code>luagit2_config</code>	None
<code>entry_name</code> (string)	
<code>value</code> (number)	

1. `luagit2_config` : The config whose entries are to be set.
 2. `entry_name` : The entry name to set.
 3. `value` : 64 bit integer type value to be set.
- Returns none or gives an error.

config_set_multivar

Can be used to set string values in a git config.

parameters	Returns
<code>luagit2_config</code>	None
<code>entry_name</code> (string)	
<code>regex</code> (string)	
<code>value</code> (string)	

1. `luagit2_config` : The config whose entries are to be set.
 2. `entry_name` : The entry name to set.
 3. `regex` : The regular expression to be used while setting variable values.
 4. `value` : string value to be set.
- Returns none or gives an error.

config_set_string

Can be used to set string values in a git config.

parameters	Returns
<code>luagit2_config</code>	None
<code>entry_name</code> (string)	
<code>value</code> (string)	

1. `luagit2_config` : The config whose entries are to be set.
 2. `entry_name` : The entry name to set.
 3. `value` : string value to be set.
- Returns none or gives an error.

config_snapshot

parameters	Returns
luagit2_config (parent)	luagit2_config

1. `luagit2_config`: The parent config whose snapshot is to be created.

- returns snapshot `luagit2_config` for given config or an error.

Note: This method only creates a snapshot. Only snapshot config as returned by this function is to be used while reading values. Setting values can not be performed on config snapshot.

config_free

parameters	Returns
luagit2_config	None

1. `luagit2_config`: The config to free.

- returns none or an error.

config_add_file_ondisk

parameters	Returns
luagit2_config	None
path (string)	
luagit2_config_level_t	
luagit2_repository	
force (int)	

1. `luagit2_config`: The config to add on disk.

2. `path`: The path where to save.

3. `luagit2_config_level_t`: The config level used to save the file.

4. `luagit2_repository`: The repository to save.

5. `force`: integer value fo boolean to force create a config.

- Returns None or an error.

config_new

parameters	Returns
None	luagit2_config

- No parameters required. returns an empty `luagit2_config` or an error.

config_iterator_new

parameters	Returns
luagit2_config	luagit2_config_iterator

1. `luagit2_config`: The config for which an iterator is to be created.
 - returns `luagit2_config_iterator` for the given config or an error.

config_entry_free

parameters	Returns
luagit2_config_entry	None

1. `luagit2_config_entry`: The `config_entry` to free.
 - returns `none` or an error.

config_get_entry

parameters	Returns
luagit2_config	luagit2_config_entry
entry_name (string)	

1. `luagit2_config`: The config whose entries are to be looked up.
 2. `entry_name`: The entry whose value is to be found.
- ..note:: The config entry values can be read using helper methods `config_entry_name()` and `config_entry_value()`.
- Returns `luagit2_config_entry` for that or gives an error.

config_next

parameters	Returns
luagit2_config_iterator	luagit2_config_entry

1. `luagit2_config_iterator`: The `config_iterator` to lookup values in.
 - returns next `luagit2_config_entry` or an error.
- ..note:: The config entry values can be read using helper methods `config_entry_name()` and `config_entry_value()`.

config_iterator_free

parameters	Returns
luagit2_config_iterator	None

1. `luagit2_config_iterator`: The `config_iterator` to free.
 - returns `none` or an error.

1.2.10 Cred

These methods help to create different type of Credential objects for a git repository.

Contents

- *cred_default_new*
- *cred_ssh_key_from_agent*
- *cred_ssh_key_memory_new*
- *cred_ssh_key_new*
- *cred_username_new*
- *cred_userpass_plaintext_new*
- *cred_free*

cred_default_new

parameters	Returns
None	luagit2_cred_object

- There are no parameters required. It finds default git credential details and returns `luagit2_cred_object` consisting of that data or gives an error.

cred_ssh_key_from_agent

parameters	Returns
username(string)	luagit2_cred_object

1. `username` : The Agent username required to make a credential object.
- Returns `luagit2_cred_object` for the given username or gives an error.

cred_ssh_key_memory_new

Create a new ssh key credential object reading the keys from memory.

parameters	Returns
username(string)	luagit2_cred_object
public_key(string)	
private_key(string)	
pass_phrase(string)	

1. `username` : The username to be used.
2. `public_key` : The public key.
3. `private_key` : The private key.
4. `pass_phrase` : The pass phrase.

- Returns corresponding `luagit2_cred_object` or an error.

`cred_ssh_key_new`

Create a new ssh key credential object .

parameters	Returns
<code>username(string)</code>	<code>luagit2_cred_object</code>
<code>public_key(string)</code>	
<code>private_key(string)</code>	
<code>pass_phrase(string)</code>	

1. `username` : The username to be used.
 2. `public_key` : The public key.
 3. `private_key` : The private key.
 4. `pass_phrase` : The pass phrase.
- Returns corresponding `luagit2_cred_object` or an error.

`cred_username_new`

parameters	Returns
<code>username(string)</code>	<code>luagit2_cred_object</code>

1. `username` : The username required to make a credential object.
- Returns `luagit2_cred_object` for the given username or gives an error.

`cred_userpass_plaintext_new`

parameters	Returns
<code>username(string)</code>	<code>luagit2_cred_object</code>
<code>password(string)</code>	

1. `username` : The username to be used.
 2. `password` : The password to be used.
- Returns `luagit2_cred_object` for these provided details or gives an error.

`cred_free`

parameters	Returns
<code>luagit2_cred_object</code>	<code>none</code>

1. `luagit2_cred_object` : The credential object to free.
- Returns `none` or gives an error.

1.2.11 Describe

These Methods allow to read values for `git describe` of a commit or a repository's worktree.

Contents

- *describe_commit*
- *describe_workdir*
- *describe_format*
- *describe_result_free*

describe_commit

parameters	Returns
luagit2_object	luagit2_describe_result

1. `luagit2_object` : The commit object which is to be described.
 - Returns `luagit2_describe_result` or gives an error. Must be freed by **`describe_result_free()`**.

describe_workdir

parameters	Returns
luagit2_repository	luagit2_describe_result

1. `luagit2_repository` : The repository which is to be described.
 - Returns `luagit2_describe_result` or gives an error. Must be freed by **`describe_result_free()`**.

describe_format

parameters	Returns
luagit2_describe_result	luagit2_buf

1. `luagit2_describe_result` : The describe result which is to be formatted into a `luagit2_buf`.
 - Returns `luagit2_buf` or gives an error.

describe_result_free

parameters	Returns
luagit2_describe_result	None

1. `luagit2_describe_result` : The describe result which is to be freed.
 - Returns None or gives an error.

1.2.12 Diff

These Methods allow generating `git diff` for added files to current index, between files of two trees and trees

Contents

- *diff_init_options*
- *diff_get_stats*
- *diff_index_to_index*
- *diff_index_to_workdir*
- *diff_is_sorted_icase*
- *diff_merge*
- *diff_num_deltas*
- *diff_stats_deletions*
- *diff_stats_files_changed*
- *diff_stats_insertions*
- *diff_stats_to_buf*
- *diff_to_buf*
- *diff_tree_to_index*
- *diff_tree_to_tree*
- *diff_tree_to_workdir*
- *diff_tree_to_workdir_with_index*

diff_init_options

Parameters	Returns
None	luagit2_diff_options

- There are no parameters required. This is an important function to initialize a `init_options` to be used in other functions of this module.

diff_get_stats

Parameters	Returns
luagit2_diff	luagit2_diff_stats

1. `luagit2_diff`: The diff whose stats are to be found.
- Returns `luagit2_diff_stats` or gives an error.

diff_index_to_index

parameters	Returns
luagit2_repository	luagit2_diff
luagit2_index(old)	
luagit2_index(new)	
luagit2_diff_options	

1. `luagit2_repository`: The repository where to look for the files.
 2. `luagit2_index`: The old index.
 3. `luagit2_index`: The new index to compare.
 4. `luagit2_diff_options`: The diff options.
- Returns `luagit2_diff` or gives an error.

diff_index_to_workdir

parameters	Returns
luagit2_repository	luagit2_diff
luagit2_index(old)	
luagit2_diff_options	

1. `luagit2_repository`: The repository where to look for the files.
 2. `luagit2_index`: The index to compare.
 3. `luagit2_diff_options`: The diff options.
- Returns `luagit2_diff` or gives an error.

diff_is_sorted_icase**diff_merge**

parameters	Returns
luagit2_diff (onto)	None
luagit2_diff (from)	

1. `luagit2_diff`: The diff on which to merge other diff results.
 2. `luagit2_diff`: The diff to merge.
- Returns `None` or gives an error.

diff_num_deltas

parameters	Returns
luagit2_diff	number_of_deltas(integer)

1. `luagit2_diff`: The diff whose number of deltas is to be found.
- Returns numerical value of number of deltas or gives an error.

diff_stats_deletions

parameters	Returns
luagit2_diff_stats	number_of_deletion (integer)

1. `luagit2_diff_stats`: The diff stats whose number of line deletions is to be found.
 - Returns numerical value of number of line deletions in a given diff or gives an error.

diff_stats_files_changed

parameters	Returns
luagit2_diff_stats	number_of_changes(integer)

1. `luagit2_diff_stats`: The diff stats whose number of file changes is to be found.
 - Returns numerical value of number of file changed in a given diff or gives an error.

diff_stats_insertions

parameters	Returns
luagit2_diff_stats	number_of_insertions (integer)

1. `luagit2_diff_stats`: The diff stats whose number of line insertions is to be found.
 - Returns numerical value of number of line insertions in a given diff or gives an error.

diff_stats_to_buf

parameters	Returns
luagit2_diff_stats	luagit2_buf
luagit2_diff_stats_format	
width (integer)	

1. `luagit2_diff_stats`: The diff stats which is to be formatted into a lua buf.
2. `luagit2_diff_stats_format`: The format in which the diff stats is to be formatted. This can be a value returned by any of these possible options in this helper method: `diff_stats_format_init()`. The options are **none**, ****full****, **short**, **number**, **summary**.
 - Returns `luagit2_buf` for the formatted diff stats or gives an error.

diff_to_buf

parameters	Returns
luagit2_diff_stats	luagit2_buf
luagit2_diff_format	

1. `luagit2_diff_stats`: The diff stats which is to be formatted into a lua buf.

2. `luagit2_diff_format`: The format in which the diff stats is to be formatted. This can be a value returned by any of these possible options in this helper method : `diff_format_init()`. The options are **patch**, **patch_header**, **raw**, **name_only**, **name_status**.

- Returns `luagit2_buf` for the formatted diff or gives an error.

diff_tree_to_index

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_diff</code>
<code>luagit2_tree</code>	
<code>luagit2_index(new)</code>	
<code>luagit2_diff_options</code>	

1. `luagit2_repository`: The repository to look files into.
2. `luagit2_tree`: The tree to compare with.
3. `luagit2_index`: The index to compare.
4. `luagit2_diff_options`: The necessary diff options.

- Returns a `luagit2_diff` or gives an error.

diff_tree_to_tree

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_diff</code>
<code>luagit2_tree(old)</code>	
<code>luagit2_tree (new)</code>	
<code>luagit2_diff_options</code>	

1. `luagit2_repository`: The repository to look files into.
2. `luagit2_tree`: The old tree to compare with.
3. `luagit2_tree`: The new tree to compare.
4. `luagit2_diff_options`: The necessary diff options.

- Returns a `luagit2_diff` or gives an error.

diff_tree_to_workdir

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_diff</code>
<code>luagit2_tree</code>	
<code>luagit2_diff_options</code>	

1. `luagit2_repository`: The repository to look files into.
2. `luagit2_tree`: The old tree to compare current workdir files with.
3. `luagit2_diff_options`: The necessary diff options.

- Returns a `luagit2_diff` or gives an error.

diff_tree_to_workdir_with_index

parameters	Returns
luagit2_repository	luagit2_diff
luagit2_tree	
luagit2_diff_options	

1. `luagit2_repository` : The repository to look files into.
2. `luagit2_tree` : The old tree to compare files with that are present in current workdir **and also in current index too**.
3. `luagit2_diff_options` : The necessary diff options.
 - Returns a `luagit2_diff` or gives an error.

1.2.13 Graph

Contents

- *[graph_ahead_behind](#)*
- *[graph_descendant_of](#)*

graph_ahead_behind

This Method help in comparing and finding number of commits in between 2 commits.

parameters	Returns
luagit2_repository	numbers_ahead (number)
luagit2_oid (current)	numbers_behind(number)
luagit2_oid(upstream)	

1. `luagit2_repository` : The repository whose commits are to be compared.
2. `luagit2_oid` : The current commit.
3. `luagit2_oid` : The other upstream commit with which to compare.
 - **Returns two numbers in order:**
 - a. number of commits current commit is ahead of upstream commit.
 - b. number of commits current commit is behind upstream commit.

Note: The upstream commit must be one of commits in the same tree as the other commit.

graph_descendant_of

Helps in checking if one commit is parent commit of other.

parameters	Returns
luagit2_repository	is_decendent (Boolean)
luagit2_oid (current)	
luagit2_oid(upstream)	

1. `luagit2_repository`: The repository whose commits are to be compared.
2. `luagit2_oid`: The current commit to check.
3. `luagit2_oid`: The ancestor commit with which to compare.
 - Returns a boolean value for whether current commit is a child commit of other or gives an error.

1.2.14 Ignore

Contents

- *ignore_add_rule*
- *ignore_clear_internal_rules*
- *ignore_path_is_ignored*

ignore_add_rule

Adds ignore rules for a given repository.

parameters	Returns
luagit2_repository	None
Rule (string)	

1. `luagit2_repository`: The repository where to add ignore rules.
2. `Rule`: The string value of rule to add.
 - Returns None or gives an error.

ignore_clear_internal_rules

Clears all ignore rules of a given repository.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository whose rules are to be cleared.
 - Returns none or gives an error

ignore_path_is_ignored

Checks if a file is ignored in a given repository.

parameters	Returns
luagit2_repository	is_ignored (Boolean)
Path (string)	

1. `luagit2_repository` : The repository where to check for the provided file.
2. `Path` : The path to the file which is to be checked.
 - Returns Boolean value or gives an error.

1.2.15 Index

These Methods allow us to read, modify, create, delete working index for a given `git repository`.

Contents

- *index_add*
- *index_add_bypath*
- *index_caps*
- *index_checksum*
- *index_conflict_add*
- *index_conflict_cleanup*
- *index_entry_is_conflict*
- *index_entry_stage*
- *index_entrycount*
- *index_find*
- *index_find_prefix*
- *index_get_byindex*
- *index_get_bypath*
- *index_has_conflicts*
- *index_open*
- *index_owner*
- *index_path*
- *index_read*
- *index_read_tree*
- *index_remove*
- *index_remove_bypath*
- *index_remove_directory*
- *index_set_caps*
- *index_set_version*

- *index_version*
- *index_write*
- *index_write_tree*
- *index_write_tree_to*
- *index_free*

index_add

Add an index entry to a present index in a git repo.

Parameters	Returns
luagit2_index	None
luagit2_index_entry	

1. `luagit2_index`: The index where to add a new index entry.
2. `luagit2_index_entry`: The Index entry to be added.
 - returns None or gives an error.

index_add_bypath

Add a file to a present index in a git repo.

Parameters	Returns
luagit2_index	None
file_path (string)	

1. `luagit2_index`: The index where to add a new index entry.
2. `file_path`: The path of file to be added.
 - returns None or gives an error.

index_caps

Get index capabilities flags value for a present index in a git repo.

Parameters	Returns
luagit2_index	Caps_val (integer)

1. `luagit2_index`: The index where to add a new index entry.
 - returns an integer or gives an error.

index_checksum

Get the checksum of the index. A `luagit2_oid` is returned whose str value is equal to index's checksum.

Parameters	Returns
luagit2_index	luagit2_oid

1. `luagit2_index`: The index whose checksum to find.

- returns corresponding `luagit2_oid` or gives an error.

index_conflict_add

Add or update index entries to represent a conflict. Any staged entries that exist at the given paths will be removed.

parameters	Returns
<code>luagit2_index</code>	None
<code>luagit2_index_entry(ancestor)</code>	
<code>luagit2_index_entry(ours)</code>	
<code>luagit2_index_entry(theirs)</code>	

1. `luagit2_index`: The index where to add.

2. `luagit2_index_entry`: The entry data for the ancestor of the conflict.

3. `luagit2_index_entry`: The entry data for our side of the merge conflict.

4. `luagit2_index_entry`: The entry data for their side of the merge conflict.

- Returns None or gives an error.

index_conflict_cleanup

Remove all conflicts in the index

Parameters	Returns
<code>luagit2_index</code>	None

1. `luagit2_index`: The index whose conflicts are to be cleaned up.

- returns None or gives an error.

index_entry_is_conflict

Return whether the given index entry is a conflict.

Parameters	Returns
<code>luagit2_index_entry</code>	<code>is_conflict</code> (Boolean)

1. `luagit2_index_entry`: The index entry to check.

- returns Boolean for accordingly as if the entry is a conflict or not or gives an error.

index_entry_stage

Get the staged index value for the index entry.

Parameters	Returns
<code>luagit2_index_entry</code>	<code>stage_index</code> (integer)

1. `luagit2_index_entry`: The index entry for which the stage index value is to be found.

- returns stage index value of index entry or gives an error.

index_entrycount

Get the number of entries in a `luagit2_index`.

Parameters	Returns
<code>luagit2_index</code>	<code>entry_count</code> (integer)

1. `luagit2_index`: The index whose entry count is to be found.
 - Returns integer value of entry count or gives an error.

index_find

Find the first position of any entries which point to given path in the Git index.

Parameters	Returns
<code>luagit2_index</code>	<code>idx_pos</code> (integer)
<code>file_path</code> (string)	

1. `luagit2_index`: The index where to search in.
2. `file_path`: The file path to search for.
 - Returns an integer value for the index position or gives an error

index_find_prefix

Find the first position of any entries which point to given prefix name in the Git index.

Parameters	Returns
<code>luagit2_index</code>	<code>idx_pos</code> (integer)
<code>prefix_name</code> (string)	

1. `luagit2_index`: The index where to search in.
2. `prefix_name`: The prefix name to search for.
 - Returns an integer value for the index position or gives an error.

index_get_byindex

Get an Index entry by its index number in a given git index.

Parameters	Returns
<code>luagit2_index</code>	<code>luagit2_index_entry</code>
<code>idx_num</code> (integer)	

1. `luagit2_index`: The index where to look for entry.
2. `idx_pos`: The index position of entry in the `luagit2_index`.
 - Returns corresponding `luagit2_index_entry` or gives an error.

index_get_bypath

Parameters	Returns
luagit2_index	luagit2_index_entry
file_path (string)	
stage_val (integer)	

1. `luagit2_index`: The index where to search in.
 2. `file_path`: The file path to search for.
 3. `stage_val`: The stage value for the indexentry .
- Returns corresponding `luagit2_index_entry` or gives an error.

index_has_conflicts

Check if a `luagit2_index` has conflicting entries.

Parameters	Returns
luagit2_index	has_conflicts (Boolean)

1. `luagit2_index`: The index which is to be checked.
- Returns Boolean value if index has conflicts or gives an error.

index_open

Open a git index from its path.

Parameters	Returns
index_path (string)	luagit2_index

- Returns `luagit2_index` or gives an error.

index_owner

Get the owner repo for a given git index

Parameters	Returns
luagit2_index	luagit2_reposiotory

- Returns corresponding owner `luagit2_reposiotory` or gives an error.

index_path

Get the file path of a given git index.

Parameters	Returns
luagit2_index	path (string)

- Returns corresponding path or gives an error.

index_read

Update the index by reading values from disk. (actual values in the repo)

Parameters	Returns
luagit2_index	None
force (integer)	

1. `luagit2_index`: The index to be updated.
2. `force`: integer equivalent of bool to force reading values from disk.
 - Returns none but updated values in the `luagit2_index`.

index_read_tree

Read values from a given tree into the git index.

Parameters	Returns
luagit2_index	None
luagit2_tree	

1. `luagit2_index`: The index to read values into.
2. `luagit2_tree`: The tree where to read values from.
 - Returns none but updated values in the `luagit2_index`.

index_remove

Remove an entry from index. This should remove the staged index entry.

Parameters	Returns
luagit2_index	None
file_path (string)	
stage_val (integer)	

1. `luagit2_index`: The index where to search in.
2. `file_path`: The file path to search for.
3. `stage_val`: The stage value for the index entry .
 - Returns None or gives an error.

index_remove_bypath

Remove an entry from index using only file path. This should remove the index entry completely.

Parameters	Returns
luagit2_index	None
file_path (string)	

1. `luagit2_index`: The index where to search in.
2. `file_path`: The file path to search for.

- Returns None or gives an error.

index_remove_directory

Remove an Directory from index. This should remove the staged index entry.

Parameters	Returns
luagit2_index	None
directory_path (string)	
stage_val (integer)	

1. `luagit2_index` : The index where to search in.
2. `directory_path` : The directory path to search for.
3. `stage_val` : The stage value for the index entry .

- Returns None or gives an error.

index_set_caps

Parameters	Returns
luagit2_index	None
caps_val (integer)	

1. `luagit2_index` : The index value whose capabilities flags are to be set.
2. `caps_val` : The integer value for the flags value of capabilities for index.

- Returns None or gives an error.

index_set_version

Parameters	Returns
luagit2_index	None
version_number (integer)	

1. `luagit2_index` : The index value whose capabilities flags are to be set.
2. `version_number` : The integer value of version for index.

- Returns None or gives an error.

index_version

Parameters	Returns
luagit2_index	version_number (integer)

1. `luagit2_index` : The index whose version is to be found.

- returns None or gives an error.

index_write

Write the index to disk.

Parameters	Returns
luagit2_index	None

1. `luagit2_index`: The index to be written to disk.
 - returns None or gives an error.

index_write_tree

Write index to a tree.

Parameters	Returns
luagit2_index	luagit2_oid

1. `luagit2_index`: The index to be written to tree.
 - returns `luagit2_oid` of the tree or gives an error.

index_write_tree_to

Parameters	Returns
luagit2_index	luagit2_oid
luagit2_reposiotory	

1. `luagit2_index`: The index to be written to tree.
2. `luagit2_reposiotory`: The repository to write to.
 - returns `luagit2_oid` of the tree or gives an error.

index_free

Parameters	Returns
luagit2_index	None

1. `luagit2_index`: The index to be freed.
 - returns None or gives an error.

1.2.16 Libgit

These are the most important set of functions that allow setting and reading global values for all of the libgit and thus luagit2 too.

Contents

- *init*

- *shutdown*
- *libgit2_features*
- *get_mwindow_size*
- *set_mwindow_size*
- *get_mwindow_mapped_limit*
- *set_mwindow_mapped_limit*
- *get_search_path*
- *set_search_path*
- *get_cached_memory*
- *set_cache_object_limit*
- *set_cache_max_size*
- *enable_caching*
- *get_template_path*
- *set_template_path*
- *set_user_agent*
- *get_windows_sharemode*
- *set_windows_sharemode*
- *enable_strict_object_creation*
- *enable_strict_symbolic_ref_creation*
- *enable_ofs_delta*
- *set_ssl_ciphers*
- *enable_fsync_gitdir*
- *enable_strict_hash_verification*

init

Sets up the global state and threading values for the libgit2.

- There are no params required, none are returned.

Note: This is the First function to be called before doing anything or calling other functions of luagit2. This may be called any number of times.

shutdown

Clean up the global state and threading context after calling it as many times as **init()** was called.

- There are no params required, none are returned.

libgit2_features

parameters	Returns
None	integer_value

- returns a Integer value for Query compile time options for libgit2.

get_mwindow_size

parameters	Returns
None	integer_value

- returns an Integer valued maximum mmap window size.

set_mwindow_size

parameters	Returns
integer_value	None

- Sets the maximum mmap window size for libgit2.

get_mwindow_mapped_limit

parameters	Returns
None	integer_value

- returns an Integer value of the maximum memory that will be mapped in total by the library.

set_mwindow_mapped_limit

parameters	Returns
integer_value	None

- Sets the the maximum memory that will be mapped in total by the library.

get_search_path

parameters	Returns
(string) search_path_lvl	(string) path

1. `search_path_lvl` : The search path level. can be any of **system**, **global**, **xdg**, **programdata** .

- returns String value of search path for a selected level for a config data.

set_search_path

parameters	Returns
(string) search_path_lvl	None
(string) path	

1. `search_path_lvl` : The search path level. can be any of **system**, **global**, **xdg**, **programdata** .
2. `path` : The string value of path to set.
 - Sets String value of search path for a selected level for a config data.

get_cached_memory

parameters	Returns
None	(integer_value) current
	(integer_value) allowed

- returns cache byte values.
 - a. current size.
 - b. max allowed size.

set_cache_object_limit

parameters	Returns
luagit2_otype	None
(integer) limit	

1. `luagit2_otype` : the object type whose cache size is to be set.
2. `limit` : The numerical value of cache size to be allowed for that object type.
 - Returns none or gives an error.

set_cache_max_size

parameters	Returns
integer_value	None

- Set the max value of cache size for any of the object.

enable_caching

parameters	Returns
integer_value	None

integer_value :

- a. `1` -> enable caching.
- b. `0` -> disable caching.

get_template_path

parameters	Returns
None	luagit2_buf

- there are no params required. luagit2_buf for used template path is returned.

set_template_path

parameters	Returns
(String) path	None

- Set the default template path.

set_user_agent

parameters	Returns
(String) user_agent	None

- Sets the user agent using it's name.

get_windows_sharemode

parameters	Returns
None	integer_value

- Get the numerical value of file share mode in windows systems.

Note: This methods should be used in windows systems only. Otherwise it may give errors or wrong results.

set_windows_sharemode

parameters	Returns
(Integer) share_mode	None

- Set the file share mode to be used by libgit in windows system.

Note: This methods should be used in windows systems only. Otherwise it may give errors or wrong results.

enable_strict_object_creation

parameters	Returns
integer_value	None

integer_value :

- a. *1* -> enable Strict_object_creation.
- b. *0* -> disable Strict_object_creation.
- Returns none.

enable_strict_symbolic_ref_creation

parameters	Returns
integer_value	None

integer_value :

- a. *1* -> enable strict_symbolic_ref_creation.
- b. *0* -> disable strict_symbolic_ref_creation.
- Returns none.

enable_ofs_delta

parameters	Returns
integer_value	None

integer_value :

- a. *1* -> enable ofs_delta.
- b. *0* -> disable ofs_delta.
- Returns none.

set_ssl_ciphers

parameters	Returns
(string) ciphers	None

1. Ciphers : The string value of ciphers to set.
- returns None.

enable_fsync_gitdir

parameters	Returns
integer_value	None

integer_value :

- a. *1* -> Enable synchronized writing of files in the gitdir using *fsync*.
- b. *0* -> disable synchronized writing of files in the gitdir using *fsync*.
- Returns none.

enable_strict_hash_verification

parameters	Returns
integer_value	None

integer_value :

- a. *1* -> enable strict_hash_verification.
- b. *0* -> disable strict_hash_verification.
- Returns none.

1.2.17 Note

These methods allow creating and reading git notes for git objects.

Contents

- *note_author*
- *note_committer*
- *note_commit_create*
- *note_create*
- *note_free*
- *note_message*
- *note_id*
- *note_commit_remove*
- *note_remove*
- *note_read*
- *note_commit_read*

note_author

parameters	Returns
luagit2_note	luagit2_signature

1. `luagit2_note` : The note whose author signature is to be found.
- Returns `luagit2_signature` or gives an error.

note_committer

parameters	Returns
luagit2_note	luagit2_signature

1. `luagit2_note` : The note whose committer signature is to be found.

- Returns `luagit2_signature` or gives an error.

note_commit_create

Create a note with a new commit for an object.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_oid (commit_id)</code> <code>luagit2_oid (blob_id)</code>
<code>luagit2_commit (immediate parent commit)</code>	
<code>luagit2_signature(author)</code>	
<code>luagit2_signature(committer)</code>	
<code>luagit2_oid</code>	
<code>message (string)</code>	
<code>force(int)</code>	

1. `luagit2_repository` : The repos where to make a new note.
2. `luagit2_commit` : The parent commit using which new note' creation commit will be made.
3. `luagit2_signature` : The author signature.
4. `luagit2_signature` : The committer signature.
5. `luagit2_oid` : The target object's oid.
6. `message` : The note message.
7. `force` : integer equivalent of boolean to force create note.

- **Returns these values or gives an error.**

- a. Commit's oid
- b. Note blob's oid.

note_create

Create a note without a new commit for an object.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_oid (commit_id)</code> <code>luagit2_oid (blob_id)</code>
<code>notes_ref (string)</code>	
<code>luagit2_signature(author)</code>	
<code>luagit2_signature(committer)</code>	
<code>luagit2_oid</code>	
<code>message (string)</code>	
<code>force(int)</code>	

1. `luagit2_repository` : The repos where to make a new note.
2. `notes_ref` : The note reference where new note will be created.
3. `luagit2_signature` : The author signature.
4. `luagit2_signature` : The committer signature.
5. `luagit2_oid` : The target object's oid.
6. `message` : The note message.

7. `force` : integer equivalent of boolean to force create note.

- **Returns these values or gives an error.**
 - a. Commit's oid
 - b. Note blob's oid.

note_free

parameters	Returns
luagit2_note	None

1. `luagit2_note` : The note to free.

- Returns none or gives an error.

note_message

parameters	Returns
luagit2_note	message (string)

1. `luagit2_note` : The note whose message is to be found.

- Returns message or gives an error.

note_id

parameters	Returns
luagit2_note	luagit2_oid

1. `luagit2_note` : The note whose oid is to be found.

- Returns corresponding `luagit2_oid` or gives an error.

note_commit_remove

Remove an existing note with a new commit.

parameters	Returns
luagit2_repository	luagit2_oid (commit_id)
luagit2_commit (immediate parent commit)	luagit2_oid (blob_id)
luagit2_signature(author)	
luagit2_signature(committer)	
luagit2_oid	

1. `luagit2_repository` : The repos where to make a new note.
2. `luagit2_commit` : The parent commit using which note deletion commit will be made.
3. `luagit2_signature` : The author signature.
4. `luagit2_signature` : The committer signature.

5. `luagit2_oid`: The note oid to remove.

- Returns none or gives an error.

note_remove

Remove a commit without making a new commit.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_oid</code> (commit_id)
<code>notes_ref</code> (string)	<code>luagit2_oid</code> (blob_id)
<code>luagit2_signature</code> (author)	
<code>luagit2_signature</code> (committer)	
<code>luagit2_oid</code>	

1. `luagit2_repository`: The repos where to make a new note.

2. `notes_ref`: The ref where to look for the note to be deleted.

3. `luagit2_signature`: The author signature.

4. `luagit2_signature`: The committer signature.

5. `luagit2_oid`: The note oid to remove.

- Returns none or gives an error.

note_read

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_note</code>
<code>notes_ref</code> (string)	
<code>luagit2_oid</code>	

1. `luagit2_repository`: Thhe repository where to look for notes.

2. `notes_ref`: The reference name for the notes refs.

3. `luagit2_oid`: The oid of note to be found.

- Returns corresponding `luagit2_note` or gives an error.

note_commit_read

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_note</code>
<code>luagit2_commit</code>	
<code>luagit2_oid</code>	

1. `luagit2_repository`: Thhe repository where to look for notes.

2. `luagit2_commit`: The commit corresponding to creation of note for the object.

3. `luagit2_oid`: The oid of note to be found.

- Returns corresponding `luagit2_note` or gives an error.

1.2.18 Object

These methods allow finding details for `git` objects.

Contents

- *object__size*
- *object_free*
- *object_id*
- *object_lookup*
- *object_lookup_bypath*
- *object_lookup_prefix*
- *object_owner*
- *object_short_id*
- *object_string2type*
- *object_type*
- *object_type2string*
- *object_typeisloose*

object__size

Get the object size for different **object types**.

parameters	Returns
luagit2_otype	(Number) size

1. `luagit2_otype` : The object type whose size is to be found.

- returns numerical value of size for that object type.

object_free

Free the object.

parameters	Returns
luagit2_object	None

1. `luagit2_object` : The object to free.

- returns none or gives an error.

object_id

Get the oid of given `luagit2_object`.

parameters	Returns
luagit2_object	luagit2_oid

1. `luagit2_object` : The object whose oid is to be found.

- returns corresponding `luagit2_oid` or gives an error.

object_lookup

parameters	Returns
luagit2_repository	luagit2_object
luagit2_oid	
luagit2_otype	

1. `luagit2_repository` : The repository where to look for object.

2. `luagit2_oid` : The oid of requested object.

3. `luagit2_otype` : The object type of requested object.

- Returns corresponding `luagit2_object` or gives an error.

object_lookup_bypath

parameters	Returns
luagit2_repository	luagit2_object
path	
luagit2_otype	

1. `luagit2_repository` : The repository where to look for object.

2. `path` : The path where to find objects. example : `repo_path/.git/objects`

3. `luagit2_otype` : The object type of requested object.

- Returns corresponding `luagit2_object` or gives an error.

object_lookup_prefix

parameters	Returns
luagit2_repository	luagit2_object
luagit2_oid	
length (int)	
luagit2_otype	

1. `luagit2_repository` : The repository where to look for object.

2. `luagit2_oid` : The oid of requested object.

3. `length` : The length of oid to use look to the object in repository.

4. `luagit2_otype` : The object type of requested object.

- Returns corresponding `luagit2_object` or gives an error.

object_owner

Get the owner repository of a `luagit2_object`.

parameters	Returns
<code>luagit2_object</code>	<code>luagit2_repository</code>

1. `luagit2_object` : The object whose owner repository is to be found.
 - Returns `luagit2_repository` or gives an error.

object_short_id

Get the short oid for object.

parameters	Returns
<code>luagit2_object</code>	<code>luagit2_buf</code>

1. `luagit2_object` : The object whose short oid is to be found.
 - Returns `luagit2_buf` that contains the shortened oid or gives an error.

object_string2type

Get a `luagit2_otype` from string `type_name` .

parameters	Returns
<code>type_name</code> (string)	<code>luagit2_otype</code>

1. `type_name` : The name of type to get a `luagit2_otype` for. can be any of **commit**, **blob** , **tag**, **tree** etc.
 - Returns corresponding `luagit2_otype` or gives an error.

object_type

Get object's type.

parameters	Returns
<code>luagit2_object</code>	<code>luagit2_otype</code>

1. `luagit2_object` : The object whose object type is to be found.
 - Returns `luagit2_otype` for the object's type or gives an error.

object_type2string

Get the string name of `object_type`.

parameters	Returns
<code>luagit2_otype</code>	<code>type_name(string)</code>

1. `luagit2_otype` : The object type whose string type name is to be found.
 - Returns string value of `type_name` for the object's type or gives an error.

object_typeisloose

Check if the object type is a valid loose object type.

parameters	Returns
luagit2_otype	is_loose (boolean)

1. `luagit2_otype`: The object type to check.
 - Returns boolean if the object is a valid loose object type.

1.2.19 Object Database (ODB)

These methods allow functions related to finding details about `git objects` in the repository's object database.

Contents

- *odb_exists*
- *odb_free*
- *odb_hash*
- *odb_hashfile*
- *odb_object_free*
- *odb_object_id*
- *odb_object_size*
- *odb_object_type*
- *odb_open*
- *odb_read*
- *odb_refresh*
- *odb_write*

odb_exists

parameters	Returns
luagit2_odb	exists (boolean)
luagit2_oid	

1. `luagit2_odb`: The object data base where to check.
2. `luagit2_oid`: The oid of object to check.
 - Returns boolean if the object exists in the given odb.

odb_free

parameters	Returns
luagit2_odb	None

1. `luagit2_odb` : The object to free.

- Returns none or gives an error.

odb_hash

parameters	Returns
data (string)	luagit2_oid
luagit2_otype	

1. `Data` : The string data whose hash is to be found out.

2. `luagit2_otype` : The object type for the data passed.

- Returns corresponding `luagit2_oid` for data passed or gives an error.

odb_hashfile

parameters	Returns
path (string)	luagit2_oid
luagit2_otype	

1. `path` : The file's path whose hash is to be found out.

2. `luagit2_otype` : The object type for the file passed.

- Returns corresponding `luagit2_oid` for file passed or gives an error.

odb_object_free

parameters	Returns
luagit2_odb_object	None

1. `luagit2_odb_object` : The `odb_object` to free.

- Returns none or gives an error.

odb_object_id

parameters	Returns
luagit2_odb_object	luagit2_oid

1. `luagit2_odb_object` : The `odb_object` whose oid is to be found.

- Returns corresponding `luagit2_oid` or gives an error.

odb_object_size

parameters	Returns
luagit2_odb_object	Size (number)

1. `luagit2_odb_object` : The `odb_object` whose size is to be found.
 - Returns corresponding numerical value of size or gives an error.

odb_object_type

parameters	Returns
luagit2_odb_object	luagit2_otype

1. `luagit2_odb_object` : The `odb_object` whose object type to be found.
 - Returns corresponding `luagit2_otype` or gives an error.

odb_open

parameters	Returns
path (string)	luagit2_odb

1. `path` : The path to objects for the repository. example : `repo/path/.git/objects`
 - Returns corresponding `luagit2_odb` or gives an error.

odb_read

Get a `luagit2_odb_object` from the `object_database` using it's oid.

parameters	Returns
luagit2_odb	luagit2_odb_object
luagit2_oid	

1. `luagit2_odb` : The `odb` from where to read data.
2. `luagit2_oid` : The oid of object to read.
 - Returns `luagit2_odb_object` or gives an error.

odb_refresh

parameters	Returns
luagit2_odb	None

1. `luagit2_odb` : The object to refresh.
 - Returns none or gives an error.

odb_write

Create a `luagit2_odb_object` in the `object_database`.

parameters	Returns
<code>luagit2_odb</code>	<code>luagit2_oid</code>
Buffer (string)	
<code>luagit2_otype</code>	

1. `luagit2_odb` : The odb from where to add new ofb object data.
2. Buffer : The string data to add to odb.
3. `luagit2_otype` : The type of object the data refers to.
 - Returns `luagit2_oid` of new created object in odb or gives an error.

1.2.20 Reference

These methods help in modifying , creating and reading values of `git references` for a git repository. These references are internal to working of git and are available at `.git/refs` .

Contents

- *reference_create*
- *reference_create_matching*
- *reference_dup*
- *reference_dwim*
- *reference_ensure_log*
- *reference_has_log*
- *reference_is_branch*
- *reference_is_note*
- *reference_is_remote*
- *reference_is_tag*
- *reference_is_valid_name*
- *reference_iterator_glob_new*
- *reference_iterator_new*
- *reference_list*
- *reference_lookup*
- *reference_name*
- *reference_name_to_id*
- *reference_next*
- *reference_next_name*
- *reference_owner*

- *reference_peel*
- *reference_remove*
- *reference_rename*
- *reference_resolve*
- *reference_set_target*
- *reference_symbolic_create*
- *reference_symbolic_create_matching*
- *reference_symbolic_set_target*
- *reference_symbolic_target*
- *reference_target*
- *reference_target_peel*
- *reference_type*
- *reference_free*
- *reference_iterator_free*

reference_create

Create a new reference.

parameters	Returns
luagit2_repository	luagit2_reference
name (string)	
luagit2_oid	
force (integer)	
log_message (string)	

1. `luagit2_repository`: The repository where to create a new reference.
2. `name`: The name for new git reference.
3. `luagit2_oid`: The object id pointed to by the reference.
4. `force`: Integer equivalent of bool to force create the git reference.
5. `log_message`: The message to be added to log.
 - Returns Created `luagit2_reference` or gives an error.

reference_create_matching

Conditionally create new direct reference. This methods updated the git ref according to changed object. It takes an old git reference and returns a new git reference with updated values.

parameters	Returns
luagit2_repository	luagit2_reference
name (string)	
luagit2_oid	
force (integer)	
luagit2_oid(current_id ref)	
log_message (string)	

1. `luagit2_repository` : The repository where to create a new reference.
2. `name` : The name for new git reference.
3. `luagit2_oid` : The object id pointed to by the reference.
4. `force` : Integer equivalent of bool to force create the git reference.
5. `luagit2_oid` : The current oid of git reference to be updated.
6. `log_message` : The message to be added to log.
 - Returns Created `luagit2_reference` or gives an error.

reference_dup

Get a duplicate git reference of a given git reference.

parameters	Returns
luagit2_reference	luagit2_reference

1. `luagit2_reference` : The reference whose duplicate is to be created.
 - Returns Created `luagit2_reference` or gives an error.

Note: It is very important to note that simply giving a variable the value of an old `luagit2_reference` object won't help. Thus this method becomes useful if you want multiple instances for a git reference.

reference_dwim

Lookup a reference by DWIMing its short name.

parameters	Returns
luagit2_repository	luagit2_reference
reference_name(string)	

1. `luagit2_repository` : The repository where to look for reference.
2. `reference_name` : The name of reference to look for.
 - Returns Created `luagit2_reference` or gives an error.

reference_ensure_log

Ensure there is a relog for a particular reference.

parameters	Returns
luagit2_repository	has_log (boolean)
reference_name(string)	

1. `luagit2_repository` : The repository where to look for reference's log.
2. `reference_name` : The name of reference to look for.
 - Returns Boolean or gives an error.

reference_has_log

Check if a relog exists for the specified reference.

parameters	Returns
luagit2_repository	has_log (boolean)
reference_name(string)	

1. `luagit2_repository` : The repository where to look for reference's log.
2. `reference_name` : The name of reference to look for.
 - Returns Boolean or gives an error.

reference_is_branch

Check if a reference points to a git branch.

parameters	Returns
luagit2_reference	is_branch (boolean)

1. `luagit2_reference` : The reference to check.
 - Returns Boolean or gives an error.

reference_is_note

Check if a reference points to a git note.

parameters	Returns
luagit2_reference	is_note (boolean)

1. `luagit2_reference` : The reference to check.
 - Returns Boolean or gives an error.

reference_is_remote

Check if a reference points to a git remote.

parameters	Returns
luagit2_reference	is_remote (boolean)

1. `luagit2_reference` : The reference to check.

- Returns Boolean or gives an error.

reference_is_tag

Check if a reference points to a git tag.

parameters	Returns
<code>luagit2_reference</code>	<code>is_branch</code> (boolean)

1. `luagit2_reference` : The reference to check.

- Returns Boolean or gives an error.

reference_is_valid_name

Check if a reference name is valid.

parameters	Returns
<code>luagit2_reference</code>	<code>is_valid_name</code> (boolean)

1. `luagit2_reference` : The reference to check.

- Returns Boolean or gives an error.

reference_iterator_glob_new

Get a reference iterator for references that match the given glob.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_reference_iterator</code>
<code>glob_name</code> (string)	

1. `luagit2_repository` : The repository to look in the references.

2. `glob_name` : The name of glob to be used.

- Returns corresponding `luagit2_reference_iterator` or gives an error.

reference_iterator_new

Get a reference iterator for the whole repository.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_reference_iterator</code>

1. `luagit2_repository` : The repository to look in the references.

- Returns corresponding `luagit2_reference_iterator` or gives an error.

reference_list

parameters	Returns
luagit2_repository	luagit2_strarray

1. `luagit2_repository`: The repository to look in the references.
 - Returns `luagit2_strarray` as a list of string values for git references or gives an error.

reference_lookup

Get a `luagit2_reference` from a ref name in repository.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_reference</code>
<code>reference_name(string)</code>	

1. `luagit2_repository`: The repository to look in the references.
2. `reference_name`: The name of reference to look for.
 - Returns `luagit2_reference` or gives an error.

reference_name

Get the name of a git reference.

parameters	Returns
<code>luagit2_reference</code>	<code>ref_name (string)</code>

1. `luagit2_reference`: The reference whose name is to be found.
 - Returns reference's name or gives an error.

reference_name_to_id

Get the oid for reference by simply using it's name.

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_oid</code>
<code>reference_name(string)</code>	

1. `luagit2_repository`: The repository to look in the references.
2. `reference_name`: The name of reference to look for.
 - Returns corresponding `luagit2_oid` or gives an error.

reference_next

Get the next entry in reference iterator.

parameters	Returns
<code>luagit2_reference_iterator</code>	<code>luagit2_reference</code>

1. `luagit2_reference_iterator` : The reference iterator to look for next reference value in.

- Returns next `luagit2_reference` or gives an error.

reference_next_name

Get the next entry's name in reference iterator.

parameters	Returns
<code>luagit2_reference_iterator</code>	<code>ref_name</code> (string)

1. `luagit2_reference_iterator` : The reference iterator to look for next reference value in.

- Returns next reference's name or gives an error.

reference_owner

Get the owner repository for a reference.

parameters	Returns
<code>luagit2_reference</code>	<code>luagit2_repository</code>

1. `luagit2_reference` : The reference whose owner repository is to be found.

- Returns corresponding owner `luagit2_repository` or gives an error.

reference_peel

Peel a reference till an object for given otype is found.

parameters	Returns
<code>luagit2_reference</code>	<code>luagit2_object</code>
<code>luagit2_otype</code>	

1. `luagit2_reference` : The reference to look for values in.

2. `luagit2_otype` : The object type to be found.

- Returns the first `luagit2_object` found in the reference that has requested otype.

reference_remove

Remove the reference from repository.

parameters	Returns
<code>luagit2_repository</code>	None
<code>reference_name(string)</code>	

1. `luagit2_repository` : The repository where to look for reference.

2. `reference_name` : The name of reference to be deleted.

- returns none or gives an error.

reference_rename

Rename a given reference. Get a new reference with changed name but same other values.

parameters	Returns
luagit2_reference	luagit2_reference
new_name (string)	
force (integer)	
log_message (string)	

1. `luagit2_reference` : The reference whose name is to be changed.
2. `new_name` : The new name of reference.
3. `force` : Integer equivalent of boolean to force create a reference.
4. `log_message` : The message to be entered to reflog.
 - Returns new reference with changed name or gives an error.

reference_resolve

Resolves values for a given git reference.

parameters	Returns
luagit2_reference	luagit2_reference

1. `luagit2_reference` : The reference whose values need to be Resolved.
 - Returns a new `luagit2_reference` or gives an error.

reference_set_target

Set the target for a given git reference.

parameters	Returns
luagit2_reference	luagit2_reference
luagit2_oid (target)	
log_message (string)	

1. `luagit2_reference` : The reference whose oid needs to be set.
2. `luagit2_oid` : The oid of target object.
3. `log_message` : The log message to be added to reflog.
 - Returns a new `luagit2_reference` with new target or gives an error.

reference_symbolic_create

Create a new symbolic reference.

parameters	Returns
luagit2_repository	luagit2_reference
name (string)	
target_name (string)	
force (integer)	
log_message (string)	

1. `luagit2_repository` : The repository where to create a new reference.
 2. `name` : The name for new git reference.
 3. `target_name` : The name of target to be pointed to by the reference.
 4. `force` : Integer equivalent of bool to force create the git reference.
 5. `log_message` : The message to be added to log.
- Returns Created `luagit2_reference` or gives an error.

reference_symbolic_create_matching

Conditionally create new direct reference. This methods updated the git ref according to changed object. It takes an old git reference and returns a new git reference with updated values.

parameters	Returns
luagit2_repository	luagit2_reference
name (string)	
luagit2_oid	
force (integer)	
luagit2_oid(current_id ref)	
log_message (string)	

1. `luagit2_repository` : The repository where to create a new reference.
 2. `name` : The name for new git reference.
 3. `target_name` : The name of target to be pointed to by the reference.
 4. `force` : Integer equivalent of bool to force create the git reference.
 5. `current` : The old value of the reference when updating .
 6. `log_message` : The message to be added to log.
- Returns Created `luagit2_reference` or gives an error.

reference_symbolic_set_target

Set the target for a given symbolic git reference.

parameters	Returns
luagit2_reference	luagit2_reference
target_name (string)	
log_message (string)	

1. `luagit2_reference` : The reference whose oid needs to be set.
2. `target_name` : The name of target object.

3. `log_message` : The log message to be added to reflog.

- Returns a new `luagit2_reference` with new target or gives an error.

reference_symbolic_target

Get the target's name of a given symbolic reference.

parameters	Returns
<code>luagit2_reference</code>	<code>target_name</code>

1. `luagit2_reference` : The reference to look for target in.

- Returns the `target_name` in the symbolic reference or gives an error.

reference_target

Get the target's oid of a given reference.

parameters	Returns
<code>luagit2_reference</code>	<code>luagit2_oid</code>

1. `luagit2_reference` : The reference to look for target in.

- Returns the `luagit2_oid` in the reference or gives an error.

reference_target_peel

Return the peeled OID target of this reference.

parameters	Returns
<code>luagit2_reference</code>	<code>luagit2_oid</code>

1. `luagit2_reference` : The reference to look for values in.

- Returns the `luagit2_oid` in the reference or gives an error.

reference_type

parameters	Returns
<code>luagit2_reference</code>	<code>luagit2_ref_type</code>

- Returns corresponding `luagit2_ref_type` or gives an error.

reference_free

parameters	Returns
<code>luagit2_reference</code>	<code>None</code>

- Frees a given `luagit2_reference`. Returns none or gives an error.

reference_iterator_free

parameters	Returns
luagit2_reference_iterator	None

- Frees a given `luagit2_reference_iterator`. Returns none or gives an error.

1.2.21 Reflog

These methods help in reading reflogs for given reference.

Contents

- *reflog_read*
- *reflog_entrycount*
- *reflog_free*
- *reflog_entry_byindex*
- *reflog_entry_committer*
- *reflog_entry_message*
- *reflog_entry_id_new*
- *reflog_entry_id_old*
- *reflog_delete*
- *reflog_drop*
- *reflog_append*

reflog_read

Read reflog for given reference.

parameters	Returns
luagit2_repository	luagit2_reflog
name	

1. `luagit2_repository` : The repository where to look for reflogs.
 2. `name` : The name for which to look for reflog. example : HEAD or master.
- Returns `luagit2_reflog` accordingly or gives an error.

reflog_entrycount

parameters	Returns
luagit2_reflog	count (number)

1. `luagit2_reflog` : The reflog whose details to find.

- Returns count of number of entries accordingly or gives an error.

reflog_free

parameters	Returns
luagit2_reflog	None

1. `luagit2_reflog`: The reflog to free.

- Returns None or gives an error.

reflog_entry_byindex

Get a reflog entry from `luagit2_reflog`.

parameters	Returns
<code>luagit2_reflog</code>	<code>luagit2_reflog_entry</code>
<code>index_number(integer)</code>	

1. `luagit2_reflog`: The reflog where to look for entries.
2. `index_number`: The index number of entry in reflog which is being looked up.

- Returns `luagit2_reflog_entry` accordingly or gives an error.

reflog_entry_committer

Find reflog entry committer sign.

parameters	Returns
<code>luagit2_reflog_entry</code>	<code>luagit2_signature</code>

1. `luagit2_reflog_entry`: The reflog entry whose committer is to be found.

- Returns committer's `luagit2_signature` accordingly or gives an error.

reflog_entry_message

Find reflog entry message.

parameters	Returns
<code>luagit2_reflog_entry</code>	<code>message(string)</code>

1. `luagit2_reflog_entry`: The reflog entry whose message is to be found.

- Returns message or gives an error.

reflog_entry_id_new

Get the reflog entry's new id.

parameters	Returns
luagit2_reflog_entry	luagit2_oid

1. `luagit2_reflog_entry`: The reflog entry whose oid is to be found.
 - Returns `luagit2_oid` or gives an error.

reflog_entry_id_old

Get the reflog entry's old id.

parameters	Returns
luagit2_reflog_entry	luagit2_oid

1. `luagit2_reflog_entry`: The reflog entry whose oid is to be found.
 - Returns `luagit2_oid` or gives an error.

reflog_delete

Delete reflog for given reference.

parameters	Returns
luagit2_repository	None
name	

1. `luagit2_repository`: The repository where to delete the reflogs.
2. `name`: The reflog which to delete. example : HEAD or master.
 - Returns None or gives an error.

reflog_drop

Drop a reflog_entrty in a reflog.

parameters	Returns
luagit2_repository	None
index_number	
rewrite_previous_entry(int)	

1. `luagit2_repository`: The repository where to drop the reflogs.
2. `index_number`: The entry's index number which to drop.
3. `rewrite_previous_entry`: the integer equivalent of boolean whether to rewrite reflog or not. o ensure there's no gap in the log history, set `rewrite_previous_entry` param value to 1. When deleting entry n, member `old_oid` of entry n-1 (if any) will be updated with the value of member `new_oid` of entry n+1.
 - Returns None or gives an error.

reflog_append

append a new entry to an existing reflog.

parameters	Returns
luagit2_reflog	None
luagit2_oid	
luagit2_signature	
Message(string)	

1. `luagit2_reflog` : The reflog where to append new entry.
2. `luagit2_oid` : The OID the reference is now pointing to.
3. `luagit2_signature` : The committer's signature.
4. `message` : the reflog entry message.
 - Returns none or gives an error.

1.2.22 Repository

Contents

- *repository_commondir*
- *repository_config*
- *repository_config_snapshot*
- *repository_detach_head*
- *repository_get_namespace*
- *repository_head*
- *repository_head_detached*
- *repository_head_for_worktree*
- *repository_head_unborn*
- *repository_ident*
- *repository_index*
- *repository_init*
- *repository_is_bare*
- *repository_is_empty*
- *repository_is_shallow*
- *repository_is_worktree*
- *repository_message*
- *repository_message_remove*
- *repository_odb*
- *repository_open*

- *repository_open_bare*
- *repository_open_from_worktree*
- *repository_path*
- *repository_refdb*
- *repository_set_head*
- *repository_set_head_detached*
- *repository_set_ident*
- *repository_set_namespace*
- *repository_set_workdir*
- *repository_state*
- *repository_state_cleanup*
- *repository_workdir*
- *repository_free*
- *repository_submodule_cache_all*
- *repository_submodule_cache_clear*
- *repository_set_index*
- *repository_set_bare*
- *repository_set_config*
- *repository_set_odb*
- *repository__cleanup*

repository_commondir

Get the path of the shared common directory for this repository.

parameters	Returns
luagit2_repository	common_dir_path (string)

1. `luagit2_repository`: The repository whose path is to be found.
 - Returns common dir path of given repository.

repository_config

Get the repository's config.

parameters	Returns
luagit2_repository	luagit2_config

1. `luagit2_repository`: The repository whose config is to be found.
 - Returns corresponding `luagit2_config` or gives an error.

repository_config_snapshot

Get the repository's config snapshot.

parameters	Returns
luagit2_repository	luagit2_config

1. `luagit2_repository`: The repository whose config is to be found.
 - Returns corresponding snapshot type of `luagit2_config` or gives an error.

repository_detach_head

Detach the repository's HEAD so that HEAD now points to nothing.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository whose HEAD is to be detached.
 - Returns None or gives an error.

repository_get_namespace

Get the namespace value for a given repository.

parameters	Returns
luagit2_repository	namespace (string)

1. `luagit2_repository`: The repository whose namespace is to be found.
 - Returns corresponding namespace or gives an error.

repository_head

Get the Reference being pointed by the repository's HEAD.

parameters	Returns
luagit2_repository	luagit2_reference

1. `luagit2_repository`: The repository whose reference being pointed by HEAD is to be found.
 - Returns corresponding `luagit2_reference` or gives an error.

repository_head_detached

Check if repository's HEAD is detached or not.

parameters	Returns
luagit2_repository	is_detached (boolean)

1. `luagit2_repository`: The repository which is to be checked.
 - Returns Boolean equivalent for if the HEAD is detached or not.

repository_head_for_worktree

Get The HEAD for the worktree.

parameters	Returns
luagit2_repository	luagit2_reference
worktree_name (string)	

1. `luagit2_repository`: The repository where to look for worktree.
 2. `worktree_name`: The name of worktree to look for.
- Returns corresponding `luagit2_reference` or gives an error.

repository_head_unborn

Check if HEAD for the repository is unborn, it is not at all created.

parameters	Returns
luagit2_repository	is_unborn (boolean)

1. `luagit2_repository`: The repository which is to be checked.
- Returns Boolean equivalent for if the HEAD is unborn or not.

repository_ident

Get the repository's identity details.

parameters	Returns
luagit2_repository	name (string)
	email (string)

1. `luagit2_repository`: The repository whose identity details are to be found.
- Returns two values as name and email for set repo's variables.

repository_index

Get the repository's currently active Index.

parameters	Returns
luagit2_repository	luagit2_index

1. `luagit2_repository`: The repository whose index is to be found.
- Returns corresponding `luagit2_index` or gives an error .

repository_init

Initialize an empty repository.

parameters	Returns
path_to_repo (string)	luagit2_repository
is_bare (integer)	

1. `path_to_repo` : The path where new repository is to be initialized.
2. `is_bare` : integer equivalent of boolean for if the new repository should be bare or not.
 - Returns a new `luagit2_repository` or gives an error.

repository_is_bare

Check if a repository is bare.

parameters	Returns
luagit2_repository	is_bare (boolean)

1. `luagit2_repository` : The repository which is to be checked.
 - Returns Boolean equivalent for repository it is bare or not.

repository_is_empty

Check if a repository is empty.

parameters	Returns
luagit2_repository	is_empty (boolean)

1. `luagit2_repository` : The repository which is to be checked.
 - Returns Boolean equivalent for if repository is empty or not.

repository_is_shallow

Check if a repository is shallow.

parameters	Returns
luagit2_repository	is_shallow (boolean)

1. `luagit2_repository` : The repository which is to be checked.
 - Returns Boolean equivalent for if repository is shallow or not.

repository_is_worktree

Check if the repository is also worktree. The answer may be different for a submodule repository.

parameters	Returns
luagit2_repository	is_worktree (boolean)

1. `luagit2_repository` : The repository which is to be checked.
 - Returns Boolean equivalent for if repository is worktree or not.

repository_message

Get the repository's message.

parameters	Returns
luagit2_repository	luagit2_buf

1. `luagit2_repository`: The repository whose message is to be found.
 - Returns message in a `luagit2_buf` or gives an error.

repository_message_remove

Remove message for the repository.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository whose message is to be removed.
 - Returns none or gives an error.

repository_odb

Get a repo's odb (object database).

parameters	Returns
luagit2_repository	luagit2_odb

1. `luagit2_repository`: The repository whose odb is to be removed.
 - Returns corresponding `luagit2_odb` or gives an error.

repository_open

Open a repository at a given path.

parameters	Returns
path_to_repo (string)	luagit2_repository

1. `path_to_repo`: The path to repository to be opened.
 - Returns corresponding `luagit2_repository` or gives an error.

repository_open_bare

Open a repository as bare repository at a given path.

parameters	Returns
path_to_repo (string)	luagit2_repository

1. `path_to_repo`: The path to repository to be opened.
 - Returns corresponding `luagit2_repository` or gives an error.

repository_open_from_worktree

parameters	Returns
luagit2_worktree	luagit2_repository

1. `luagit2_worktree` : The worktree using which repository is to be opened.
 - Returns corresponding `luagit2_repository` or gives an error.

repository_path

Get the repository's path.

parameters	Returns
luagit2_repository	path (string)

1. `luagit2_repository` : The repository whose path is to be found.
 - Returns corresponding repository's path or gives an error.

repository_refdb

Get the repository's reference database.

parameters	Returns
luagit2_repository	luagit2_refdb

1. `luagit2_repository` : The repository whose reference database is to be found .
 - Returns corresponding repository's `luagit2_refdb` or gives an error.

repository_set_head

set the HEAD of the repository to point to a given ref name.

parameters	Returns
luagit2_repository	None
ref_name (string)	

1. `luagit2_repository` : The repository whose HEAD is to be set.
2. `ref_name` : The reference name where the repository should point at.
 - Returns None or gives an error.

repository_set_head_detached

Set the detached HEAD to point to a commit. It will not make HEAD point to a reference. HEAD simply points to a given commit id.

parameters	Returns
luagit2_repository	None
luagit2_oid	

1. `luagit2_repository`: The repository whose HEAD is to be set in a detached manner.
2. `luagit2_oid`: Object id of the Commit the HEAD should point to.
 - Returns None or gives error.

repository_set_ident

parameters	Returns
<code>luagit2_repository</code>	None
<code>name (string)</code>	
<code>email(string)</code>	

1. `luagit2_repository`: The repository whose identity variables are to be set.
2. `name`: The name to be set.
3. `email`: The email variable to be set.
 - Returns none or gives an error.

repository_set_namespace

Set the namespace of the repository .

parameters	Returns
<code>luagit2_repository</code>	None
<code>namespace(string)</code>	

1. `luagit2_repository`: The repository whose namespace is to be set.
2. `namespace`: The namespace to be used.
 - Returns None or gives an error.

repository_set_workdir

Set the work directory of the repository .

parameters	Returns
<code>luagit2_repository</code>	None
<code>workdir(string)</code>	

1. `luagit2_repository`: The repository whose workdir is to be set.
2. `workdir`: The path of work directory to be used.
 - Returns None or gives an error.

repository_state

Get the integer value of a repository's state values.

parameters	Returns
<code>luagit2_repository</code>	<code>state(integer)</code>

1. `luagit2_repository`: the repository whose state values are to be found.

- Returns an integer value for state values flag set.

repository_state_cleanup

Cleanup the state values for a repository.

parameters	Returns
<code>luagit2_repository</code>	None

1. `luagit2_repository`: The repository whose state values are to be cleaned.

- Returns none or gives error.

repository_workdir

Get the repository's work directory.

parameters	Returns
<code>luagit2_repository</code>	<code>work_dir</code> (string)

1. `luagit2_repository`: The repository whose workdir is to be found.

- Returns the workdir path for the repository or gives an error.

repository_free

Frees a used `luagit2_repository` object.

parameters	Returns
<code>luagit2_repository</code>	None

1. `luagit2_repository`: The repository object to be freed.

- Returns None or gives error.

repository_submodule_cache_all

Cache all the objects of a repository's submodules.

parameters	Returns
<code>luagit2_repository</code>	None

1. `luagit2_repository`: The repository whose objects are to be cached.

- Returns none.

repository_submodule_cache_clear

Clear all the cache objects of a repository's submodules.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository whose caches are to be cleared.
- Returns none.

repository_set_index

Set the repository's current index.

parameters	Returns
luagit2_repository	None
luagit2_index	

1. `luagit2_repository`: The repository whose index is to be set.
 2. `luagit2_index`: The index to be set.
- Returns None or gives error.

repository_set_bare

Set the repo to be bare.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository to be set bare.
- Returns none.

repository_set_config

Set the repository's current config values.

parameters	Returns
luagit2_repository	None
luagit2_config	

1. `luagit2_repository`: The repository whose config is to be set.
 2. `luagit2_config`: The config to be set.
- Returns None or gives error.

repository_set_odb

Set the repository's current odb.

parameters	Returns
luagit2_repository	None
luagit2_odb	

1. `luagit2_repository`: The repository whose Odb is to be set.
 2. `luagit2_odb`: The Odb to be set.
- Returns None or gives error.

repository__cleanup

Cleans up the repository.

parameters	Returns
luagit2_repository	None

1. `luagit2_repository`: The repository to be cleaned.
- Returns none.

1.2.23 Reset

This method allows to do a `git reset` for the passed commit in the given repository.

reset

Sets the current head to the specified commit oid and resets the index and working tree to match.

parameters	Returns
luagit2_repository	None
luagit2_commit	
option (string)	

1. `luagit2_repository`: The repository where to reset the changes.
 2. `luagit2_commit`: The commit till where changes are to be reset.
 3. `option`: Specify the type of reset to be performed. This option can be any of **soft**, **mixed** or **hard**.
- Returns None or gives an error.

1.2.24 Revert

This method allows to do a `git revert` for the passed commit in the given repository.

revert

Reverts the given commit, producing changes in the index and working directory.

parameters	Returns
luagit2_repository	None
luagit2_commit	

1. `luagit2_repository`: The repository where to revert the changes.
2. `luagit2_commit`: The commit whose introduced changes are to be reverted.
 - Returns None or gives an error.

1.2.25 Revparse

These methods help in findings refs and parsing revision strings for `git` objects.

revparse

parameters	Returns
luagit2_repository	luagit2_revspec
spec (string)	

1. `luagit2_repository`: The repository from where to look for and parse the given spec string.
2. `spec`: The spec string to parse. example: `origin/master..HEAD` to get refspec data between **master** and **HEAD**.

Note: See `man gitrevisions` or http://git-scm.com/docs/git-rev-parse.html#_specifying_revisions for information on the syntax of spec string accepted.

- Returns `luagit2_revspec` which can be later used by helper methods : `revspec_from()` and `revspec_to()` to get the *From* & 'to' objects pointed by `luagit2_revspec`,

revparse_single

parameters	Returns
luagit2_repository	luagit2_object
spec (string)	

1. `luagit2_repository`: The repository from where to look for and parse the given spec string.
2. `spec`: The spec string to parse. This should point to only one of the objects.
 - Returns corresponding `luagit2_object` being pointed or gives an error.

1.2.26 Revwalk

These methods allows to do rewalk the repository's refs.

Contents

- *revwalk_new*
- *revwalk_next*
- *revwalk_push*
- *revwalk_push_ref*
- *revwalk_free*
- *revwalk_reset*
- *revwalk_push_head*
- *revwalk_repository*

revwalk_new

parameters	Returns
luagit2_repository	luagit2_revwalk

1. `luagit2_repository`: The repository those references are to be read.
 - Returns `luagit2_revwalk` for the correspondings references of given repository or gives an error.

revwalk_next

parameters	Returns
luagit2_revwalk	luagit2_oid

1. `luagit2_revwalk`: The revwalk which is to be iterated over for its entries.
 - Returns `luagit2_oid` for the next reference of given revwalk of repository or gives an error.

revwalk_push

parameters	Returns
luagit2_repository	None
luagit2_oid	

1. `luagit2_repository`: The repository where to push new refs.
2. `luagit2_oid`: The refernce's oid to push.
 - Returns none or gives an error.

revwalk_push_ref

parameters	Returns
luagit2_repository	None
ref_name(string)	

1. `luagit2_repository` : The repository where to push new refs.
 2. `ref_name` : The reference's name which is to be added.
- Returns none or gives an error.

`revwalk_free`

parameters	Returns
<code>luagit2_revwalk</code>	None

1. `luagit2_revwalk` : The revwalk which is to be freed.
- Returns None or gives an error.

`revwalk_reset`

parameters	Returns
<code>luagit2_revwalk</code>	None

1. `luagit2_revwalk` : The revwalk which is to be reset.
- Returns None or gives an error.

`revwalk_push_head`

parameters	Returns
<code>luagit2_revwalk</code>	None

1. `luagit2_revwalk` : The revwalk where to add **HEAD** of the git repository.
- Returns None or gives an error.

`revwalk_repository`

parameters	Returns
<code>luagit2_revwalk</code>	<code>luagit2_repository</code>

1. `luagit2_revwalk` : The revwalk whose owner repository is to be found.
- Returns owner `luagit2_repository` or gives an error.

1.2.27 Signature

These methods allow us to create `git signatures` to be used while creating commits or tags.

Contents

- *`signature_default`*

- *signature_dup*
- *signature_from_buffer*
- *signature_now*
- *signature_free*

signature_default

parameters	Returns
luagit2_repository	luagit2_signature

1. `luagit2_repository`: The repository's whose default signature is to be found.
 - Returns corresponding `luagit2_signature` or gives an error.

signature_dup

parameters	Returns
luagit2_signature	luagit2_signature

1. `luagit2_signature`: The old signature whose duplicate is to be made.
 - Returns corresponding duplicate `luagit2_signature` or gives an error.

signature_from_buffer

parameters	Returns
sign_data (string)	luagit2_signature

1. `sign_data`: The signature data used to make a `luagit2_signature` .
 - Returns corresponding `luagit2_signature` or gives an error.

signature_now

parameters	Returns
username (string)	luagit2_signature
password (string)	

1. `username`: The username of signature.
2. `password`: The password of signature.
 - returns `luagit2_signature` or gives an error.

signature_free

parameters	Returns
luagit2_signature	None

1. `luagit2_signature` : The signature to be freed.

- Returns None or gives an error.

1.2.28 Status

These Methods help in getting `git status`

Contents

- *status_list_new*
- *status_list_free*
- *status_list_entrycount*
- *status_byindex*

status_list_new

Get a list of status list entries for the repository. It should be freed using **status_list_free**.

Parameters	Returns
luagit2_repository	luagit2_status_list

1. `luagit2_repository` : The repo whose status list is to be found.

- Creates a corresponding `luagit2_status_list` or gives an error.

status_list_free

It frees the used **luagit2_status_list**.

Parameters	Returns
luagit2_status_list	None

1. `luagit2_status_list` : The status list to free.

- Returns none or gives an error.

status_list_entrycount

It frees the used **luagit2_status_list**.

Parameters	Returns
luagit2_status_list	num_count (ineger)

1. `luagit2_status_list` : The status list whose entry count is to be found.

- Returns number of entries in the status list or gives an error.

status_byindex

Parameters	Returns
<code>luagit2_status_list</code>	<code>luagit2_status_entry</code>
<code>index_number</code> (integer)	

1. `luagit2_status_list` : The status list where to find entry.

2. `index_number` : The index number of entry. Must be less than entry count. since the indexing starts from 0.

- Returns corresponding `luagit2_status_entry` in the status list or gives an error.

Note: Having received the status entry, one may use helper method **status_type** to find what type of entry is it. Also One may use helper method **status_file_path** to find the changed file paths for that status entry.

1.2.29 Tag

These methods help in creating and reading values of `git tags` for a given repository.

Contents

- *tag_annotation_create*
- *tag_create*
- *tag_create_frombuffer*
- *tag_create_lightweight*
- *tag_delete*
- *tag_free*
- *tag_id*
- *tag_list*
- *tag_list_match*
- *tag_lookup*
- *tag_owner*
- *tag_tagger*
- *tag_target*
- *tag_target_id*
- *tag_target_type*
- *tag_message*
- *tag_name*

tag_annotation_create

parameters	Returns
luagit2_repository	luagit2_oid
Tag_name (string)	
luagit2_object	
luagit2_signature	
tag_message (string)	

1. `luagit2_repository` : The repository in which the tag is to be created.
 2. `tag_name` : The tag name of tag to be created.
 3. `luagit2_object` : The target object targetting to which the tag should be created.
 4. `luagit2_signature` : The signature to be used to create tag.
 5. `tag_message` : The tag message.
- Returns `luagit2_oid` for the new created annotated tag.

tag_create

parameters	Returns
luagit2_repository	luagit2_oid
Tag_name (string)	
luagit2_object	
luagit2_signature	
tag_message (string)	
force (integer)	

1. `luagit2_repository` : The repository in which the tag is to be created.
 2. `tag_name` : The tag name of tag to be created.
 3. `luagit2_object` : The target object targetting to which the tag should be created.
 4. `luagit2_signature` : The signature to be used to create tag.
 5. `tag_message` : The tag message.
 6. `force` : integer equivalent for boolean whether to do a force creation.
- Returns `luagit2_oid` for the new created tag.

tag_create_frombuffer

parameters	Returns
luagit2_repository	luagit2_oid
data (string)	
force (integer)	

1. `luagit2_repository` : The repository in which the tag is to be created.
2. `data` : The data to be used to create tag.
3. `force` : integer equivalent for boolean whether to do a force creation.

- Returns `luagit2_oid` for the new created tag.

`tag_create_lightweight`

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_oid</code>
<code>Tag_name</code> (string)	
<code>luagit2_object</code>	
<code>force</code> (integer)	

1. `luagit2_repository` : The repository in which the tag is to be created.
 2. `Tag_name` : The tag name.
 3. `luagit2_object` : The target object targetting to which the tag should be created.
 4. `force` : integer equivalent for boolean whether to do a force creation.
- Returns `luagit2_oid` for the new created tag.

`tag_delete`

parameters	Returns
<code>luagit2_repository</code>	None
<code>tag_name</code>	

- Deletes a tag in a given repository. Returns none or gives an error.

`tag_free`

parameters	Returns
<code>luagit2_tag</code>	None

- Frees a tag. Returns None or gives an errors

`tag_id`

parameters	Returns
<code>luagit2_tag</code>	<code>luagit2_oid</code>

1. `luagit2_tag` : The tag whose oid is to be found.
- Returns `luagit2_oid` of the tag or gives an error.

`tag_list`

parameters	Returns
<code>luagit2_repository</code>	<code>luagit2_strarray</code>

1. `luagit2_repository` : The repository whose list of tags is to be found.
- returns `luagit2_strarray` as a list of created tags or gives an error.

tag_list_match

parameters	Returns
luagit2_repository	luagit2_strarray
pattern (string)	

1. `luagit2_repository`: The repository whose list of tags is to be found.
2. `pattern`: The pattern matching which tags will be listed out
 - returns `luagit2_strarray` as a list of created tags or gives an error.

tag_lookup

parameters	Returns
luagit2_repository	luagit2_tag
luagit2_oid	

1. `luagit2_repository`: The repository where to look for the tag.
2. `luagit2_oid`: The oid of tag to lookup.
 - returns `luagit2_tag` or gives an error.

tag_owner

parameters	Returns
luagit2_tag	luagit2_repository

1. `luagit2_tag`: tag whose owner repository is to be found.
 - Returns owner repository `luagit2_repository` or gives an error.

tag_tagger

parameters	Returns
luagit2_tag	luagit2_signature

1. `luagit2_tag`: tag whose tagger is to be found.
 - Returns tagger's signature `luagit2_signature` or gives an error.

tag_target

parameters	Returns
luagit2_tag	luagit2_object

1. `luagit2_tag`: tag whose target object is to be found.
 - Returns target `luagit2_object` or gives an error.

tag_target_id

parameters	Returns
luagit2_tag	luagit2_oid

1. `luagit2_tag`: tag whose target object's oid is to be found.
 - Returns target `luagit2_oid` or gives an error.

tag_target_type

parameters	Returns
luagit2_tag	luagit2_otype

1. `luagit2_tag`: tag whose target object type is to be found.
 - Returns target object type `luagit2_otype` or gives an error.

tag_message

parameters	Returns
luagit2_tag	message (string)

1. `luagit2_tag`: tag whose message is to be found.
 - Returns tag message or gives an error.

tag_name

parameters	Returns
luagit2_tag	name (string)

1. `luagit2_tag`: tag whose name is to be found.
 - Returns tag name or gives an error.

1.2.30 Tree

These Methods allow reading values from a `git tree` for a given repository and also helps in comparing them.

Contents

- *tree_entry_byid*
- *tree_entry_byindex*
- *tree_entry_byname*
- *tree_entry_bypath*
- *tree_entry_cmp*

- *tree_entry_filemode*
- *tree_entry_filemode_raw*
- *tree_entry_id*
- *tree_entry_name*
- *tree_entry_to_object*
- *tree_entry_type*
- *tree_entrycount*
- *tree_id*
- *tree_lookup*
- *tree_lookup_prefix*
- *tree_owner*
- *tree_entry_free*
- *tree_free*

tree_entry_byid

parameters	Returns
luagit2_tree	luagit2_tree_entry
luagit2_oid	

1. `luagit2_tree`: The tree in which entry is to be looked up.
 2. `luagit2_oid`: The tree entry oid which is being looked up.
- Returns corresponding `luagit2_tree_entry` or gives an error.

tree_entry_byindex

parameters	Returns
luagit2_tree	luagit2_tree_entry
index_number (integer)	

1. `luagit2_tree`: The tree in which entry is to be looked up.
 2. `index_number`: The tree entry index number is being looked up in the tree .
- Returns corresponding `luagit2_tree_entry` or gives an error.

tree_entry_byname

parameters	Returns
luagit2_tree	luagit2_tree_entry
file_name (string)	

1. `luagit2_tree`: The tree in which entry is to be looked up.

2. `file_name` : The file name which is being looked up in the tree .
 - Returns corresponding `luagit2_tree_entry` or gives an error.

tree_entry_bypath

parameters	Returns
<code>luagit2_tree</code>	<code>luagit2_tree_entry</code>
<code>file_path</code> (string)	

1. `luagit2_tree` : The tree in which entry is to be looked up.
2. `file_path` : The path for file which is being looked up in the tree .
 - Returns corresponding `luagit2_tree_entry` or gives an error.

tree_entry_cmp

parameters	Returns
<code>luagit2_tree_entry</code>	Integer (>0, ==0 , <0)
<code>luagit2_tree_entry</code>	

1. `luagit2_tree_entry` : The first tree entry to compare.
2. `luagit2_tree_entry` : The second tree entry to compare.
 - **Returns an integer value corresponding to**
 - a. (>0) if `entry_first` > `entry_second`.
 - b. (=0) if `entry_first` = `entry_second`.
 - c. (<0) if `entry_first` < `entry_second`.

tree_entry_filemode

parameters	Returns
<code>luagit2_tree_entry</code>	<code>luagit2_filemode</code>

1. `luagit2_tree_entry` : The tree entry whose file mode is to be found.
 - Returns corresponding `luagit2_filemode` or an error.

tree_entry_filemode_raw

parameters	Returns
<code>luagit2_tree_entry</code>	<code>luagit2_filemode</code>

1. `luagit2_tree_entry` : The tree entry whose file mode is to be found.
 - Returns corresponding raw `luagit2_filemode` or an error.

tree_entry_id

parameters	Returns
luagit2_tree_entry	luagit2_oid

1. `luagit2_tree_entry`: The tree entry whose oid is to be found.

- Returns corresponding `luagit2_oid` or an error.

tree_entry_name

parameters	Returns
luagit2_tree_entry	name (string)

1. `luagit2_tree_entry`: The tree entry whose name is to be found.

- Returns corresponding string name or an error.

tree_entry_to_object

parameters	Returns
luagit2_repository	luagit2_object
luagit2_tree_entry	

1. `luagit2_repository`: The repository where to lookup for tree entry.

2. `luagit2_tree_entry`: The tree entry which is to be converted to `luagit2_object`.

- Returns corresponding `luagit2_object` or an error.

tree_entry_type

parameters	Returns
luagit2_repository	luagit2_otype
luagit2_tree_entry	

1. `luagit2_repository`: The repository where to lookup for tree entry.

2. `luagit2_tree_entry`: The tree entry whose object type is to be found.

- Returns corresponding `luagit2_otype` or an error.

tree_entrycount

parameters	Returns
luagit2_tree	entry_count(number)

1. `luagit2_tree`: The tree whose entry count is to be found.

- Returns numerical value of number of entries or an error.

tree_id

parameters	Returns
luagit2_tree	luagit2_oid

1. `luagit2_tree` : The tree whose oid is to be found.
 - Returns corresponding `luagit2_oid` or an error.

tree_lookup

parameters	Returns
luagit2_repository	luagit2_tree
luagit2_oid	

1. `luagit2_repository` : The repository where to lookup for tree.
2. `luagit2_oid` : The oid of tree being looked up.
 - Returns corresponding `luagit2_tree` or an error.

tree_lookup_prefix

parameters	Returns
luagit2_repository	luagit2_tree
luagit2_oid	
length (int)	

1. `luagit2_repository` : The repository where to lookup for tree.
2. `luagit2_oid` : The oid of tree being looked up.
3. `length` : The length of oid to be used for looking up the tree in repository.
 - Returns corresponding `luagit2_tree` or an error.

tree_owner

parameters	Returns
luagit2_tree	luagit2_repository

1. `luagit2_tree` : The tree whose owner is to be found.
 - Returns owner `luagit2_repository` or gives an error.

tree_entry_free

parameters	Returns
luagit2_tree_entry	None

1. `luagit2_tree_entry` : The tree entry which is to be freed.
 - Returns None or gives an error

tree_free

parameters	Returns
luagit2_tree	None

1. `luagit2_tree` : The tree which is to be freed.

- Returns None or gives an error.

1.3 Examples

This section contains explanation on examples provided in `luagit2`.

1.3.1 Add Files to Index

luagit2's "add" example

(equivalent to `git add <file>`)

It shows how to modify the index

The user should pass a valid git repository's path as 1st argument. and can pass as many file path to add to the current index as he/she wishes.

Note: The passed file paths must be relative to the given repository's path

This file can be executed directly with some arguments.

```
local luagit2 = require("luagit2")

luagit2.init() -- initialize libgit2's threading and global state.

local repo_path = arg[1] -- Get the repository's path
local num_files = (#arg) -- Count total number of arguments

local repo = luagit2.repository_open(repo_path) -- open repository
local repo_index = luagit2.repository_index(repo) -- open repository's index

for i=2,num_files do
    luagit2.index_add_bypath(repo_index,arg[i]) -- Add files to index
end

luagit2.index_write(repo_index) -- Write added files to git index.

luagit2.index_free(repo_index) -- Free the used index
luagit2.repository_free(repo) -- Free used repository.

luagit2.shutdown() -- Shutdown libgit2 threading and global state.
```

1.3.2 Blame File

luagit2's "Blame" example - (equivalent to `git blame [path]`)

shows how to do a git blame on a file in a given repository.

Note that the file path passed should be accessible. It means the that the file is present in the current workdir. Errors may arise if file passed is in a different branch and is NOT in current branch.

Pass the path of repo, path of file relative to the root of the repo and the number of lines for which the blame data is to be found.

It should be noted that only one file's blame data can be seen once at a time.

```
local luagit2 = require("luagit2")
luagit2.init()

local repo_path = arg[1]
local file_path = arg[2]

local repo = luagit2.repository_open(repo_path)
local blame_file = luagit2.blame_file(repo, file_path)
-- sample functions to handle file io.
--
-- see if the file exists
function file_exists(file)
    local f = io.open(file, "rb")
    if f then f:close() end
    return f ~= nil
end

-- get all lines from a file, returns an empty
-- list/table if the file does not exist
function lines_from(file)
    if not file_exists(file) then return {} end
    lines = {}
    for line in io.lines(file) do
        lines[#lines + 1] = line
    end
    return lines
end

-- Get individual lines.
local lines = lines_from(file_path)

-- print all line numbers and the their commit details line by line.
for k,v in pairs(lines) do

    local hunk_byline = luagit2.blame_get_hunk_byline(blame_file,k)
    local sign = luagit2.blame_hunk_signature(hunk_byline)
    local _name, _email = luagit2.get_signature_details(sign)
    local last_commit_oid = luagit2.blame_hunk_commit_id(hunk_byline)
    local str_val_oid = luagit2.oid_tostr(last_commit_oid)

    print('line[' .. k .. ']', v)
    print(_name .. " " .. _email .. " " .. str_val_oid .. "\n")

end

-- You should get an output as
--
--
-- line[1]      hello
```

(continues on next page)

(continued from previous page)

```
-- test_user test@example.com 108ddee361877aa5c044d89d8dd232b8fd0f8992

-- line[2] something
-- test_user test@example.com 764e130d42fa42e1bc7a3ecb87f94ece9a6a8248

-- line[3] to tell
-- test_user test@example.com 764e130d42fa42e1bc7a3ecb87f94ece9a6a8248
--

luagit2.blame_free(blame_file)
luagit2.repository_free(repo)

luagit2.shutdown()
```

1.3.3 Clone Public Repo

luagit2's "clone" example (equivalent to git clone <repo_url> <path_where_to_clone>)

shows how clone only a public repository

The user should pass a valid git repository's remote url as 1st argument. and pass the directory path where to clone.

```
local luagit2 = require("luagit2")
luagit2.init()

if (#arg ~= 2) then
    print("2 arguments required <repo_url> <path_where_to_clone> ")
    return
end

local remote_url = arg[1]
local clone_location = arg[2]

luagit2.clone_public(remote_url,clone_location,1) -- passing 1 to print clone print_
↪progress.                                     -- pass 0 to not print clone progress.

luagit2.shutdown()
```

1.3.4 Get Diff for files in index

luagit2's "diff" example

(equivalent to git diff <file>)

shows how to get the diff data for a given repo.

The user should pass a valid git repository's path as 1st argument and 2nd argument as : The format of diff data it should print, any of : raw, patch, name_only, name_status, patch_header

Note : the passed file paths must be relative to the given repository's path

```
local luagit2 = require("luagit2")
luagit2.init()
```

(continues on next page)

(continued from previous page)

```
local repo_path = arg[1]
local format_type = arg[2]

local repo = luagit2.repository_open(repo_path)
local repo_index = luagit2.repository_index(repo)

-- Get the diff data for current index to workdir.
local diff_options = luagit2.diff_init_options()
local diff_idx_wkdir = luagit2.diff_index_to_workdir(repo, repo_index, diff_options)

local diff_format_options = luagit2.diff_format_init(format_type)

local diff_idx_wkdir_buf = luagit2.diff_to_buf(diff_idx_wkdir, diff_format_options)
local diff_data = luagit2.buf_details(diff_idx_wkdir_buf)

--Finally Print the data.
print(diff_data)

luagit2.index_free(repo_index)
luagit2.repository_free(repo)

luagit2.shutdown()
```

1.3.5 Init a blank repo and make an initial commit

luagit2's "init" example

(equivalent to git init [path] [--bare])

shows how to init a bare or non bare repository

Pass the path where to create a new repo. should be in form : <folder_name/.git>

```
local luagit2 = require("luagit2")
luagit2.init()

local repo_path = arg[1]
local is_bare = arg[2]
local repo

if is_bare == "--bare" then
    repo = luagit2.repository_init(repo_path, 1)
else repo = luagit2.repository_init(repo_path, 0)
end

-- Create an initial commit.
--
-- initializing parameters.

local repo_index = luagit2.repository_index(repo)
local index_write_tree_oid = luagit2.index_write_tree(repo_index)
local tree = luagit2.tree_lookup(repo, index_write_tree_oid)
local author_sign = luagit2.signature_default(repo)
local committer_sign = luagit2.signature_default(repo)
```

(continues on next page)

(continued from previous page)

```
--  
-- Creating an initial commit is different as there is no parent commit  
-- present so, use commit_create_initial() to make an initial commit.  
  
local new_commit_id = luagit2.commit_create_initial(repo,author_sign,  
                                                    committer_sign,"Initial commit",tree)  
  
local new_commit_str = luagit2.oid_tostr(new_commit_id)  
print("Newly created commit's id : " .. new_commit_str)  
  
-- at this point, if you run a git log in the new created  
-- repo, you should see a complete log for initial commit.  
  
luagit2.tree_free(tree)  
luagit2.index_free(repo_index)  
luagit2.repository_free(repo)  
  
luagit2.shutdown()
```