swagger-client

This documentation describes the Gitea API.

This Python package is automatically generated by the Swagger Codegen project:

- API version: 1.15
- Package version: 1.0.0
- Build package: io.swagger.codegen.languages.PythonClientCodegen

Requirements.

Python 2.7 and 3.4+

Installation & Usage

pip install

If the python package is hosted on Github, you can install directly from Github

```
pip install git+https://github.com//.git
```

(you may need to run `pip` with root permission: `sudo pip install git+https://github.com//.git`)

Then import the package:

```
import swagger_client
```

Setuptools

Install via Setuptools.

```
python setup.py install --user
```

(or `sudo python setup.py install` to install the package for all users)

Then import the package:

```
import swagger_client
```

Getting Started

Please follow the installation procedure and then run the following:

```
from __future__ import print_function
import time
import swagger_client
from swagger_client.rest import ApiException
from pprint import pprint

# Configure API key authorization: AccessToken
configuration = swagger_client.Configuration()
configuration.api_key['access_token'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['access_token'] = 'Bearer'
# Configure API key authorization: AuthorizationHeaderToken
configuration = swagger_client.Configuration()
configuration.api_key['Authorization'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['Authorization'] = 'Bearer'
```

```
# Configure HTTP basic authorization: BasicAuth
configuration = swagger_client.Configuration()
configuration.username = 'YOUR_USERNAME'
configuration.password = 'YOUR_PASSWORD'
# Configure API key authorization: SudoHeader
configuration = swagger_client.Configuration()
configuration.api_key['Sudo'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['Sudo'] = 'Bearer'
# Configure API key authorization: SudoParam
configuration = swagger_client.Configuration()
configuration.api_key['sudo'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['sudo'] = 'Bearer'
# Configure API key authorization: TOTPHeader
configuration = swagger_client.Configuration()
configuration.api_key['X-GITEA-OTP'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['X-GITEA-OTP'] = 'Bearer'
# Configure API key authorization: Token
configuration = swagger_client.Configuration()
configuration.api_key['token'] = 'YOUR_API_KEY'
# Uncomment below to setup prefix (e.g. Bearer) for API key, if needed
# configuration.api_key_prefix['token'] = 'Bearer'

# create an instance of the API class
api_instance = swagger_client.AdminApi(swagger_client.ApiClient(configuration))
owner = 'owner_example' # str | owner of the repo
repo = 'repo_example' # str | name of the repo

try:
    # Adopt unadopted files as a repository
    api_instance.admin_adopt_repository(owner, repo)
except ApiException as e:
    print("Exception when calling AdminApi->admin_adopt_repository: %s\n" % e)
```

Documentation for API Endpoints

All URIs are relative to *http://localhost/api/v1*

Class	Method	HTTP request	Description
AdminApi	**admin_adopt_repository**	**POST** /admin/unadopted/{owner}/{repo}	Adopt unadopted files as a repository
AdminApi	**admin_create_org**	**POST** /admin/users/{username}/orgs	Create an organization
AdminApi	**admin_create_public_key**	**POST** /admin/users/{username}/keys	Add a public key on behalf of a user
AdminApi	**admin_create_repo**	**POST** /admin/users/{username}/repos	Create a repository on behalf of a user
AdminApi	**admin_create_user**	**POST** /admin/users	Create a user
AdminApi	**admin_cron_list**	**GET** /admin/cron	List cron tasks
AdminApi	**admin_cron_run**	**POST** /admin/cron/{task}	Run cron task
AdminApi	**admin_delete_unadopted_repository**	**DELETE** /admin/unadopted/{owner}/{repo}	Delete unadopted files
AdminApi	**admin_delete_user**	**DELETE** /admin/users/{username}	Delete a user
AdminApi	**admin_delete_user_public_key**	**DELETE** /admin/users/{username}/keys/{id}	Delete a user's public key
AdminApi	**admin_edit_user**	**PATCH** /admin/users/{username}	Edit an existing user

Class	Method	HTTP request	Description
AdminApi	**admin_get_all_orgs**	**GET** /admin/orgs	List all organizations
AdminApi	**admin_get_all_users**	**GET** /admin/users	List all users
AdminApi	**admin_unadopted_list**	**GET** /admin/unadopted	List unadopted repositories
IssueApi	**issue_add_label**	**POST** /repos/{owner}/{repo}/issues/{index}/labels	Add a label to an issue
IssueApi	**issue_add_subscription**	**PUT** /repos/{owner}/{repo}/issues/{index}/subscriptions/{user}	Subscribe user to issue
IssueApi	**issue_add_time**	**POST** /repos/{owner}/{repo}/issues/{index}/times	Add tracked time to a issue
IssueApi	**issue_check_subscription**	**GET** /repos/{owner}/{repo}/issues/{index}/subscriptions/check	Check if user is subscribed to an issue
IssueApi	**issue_clear_labels**	**DELETE** /repos/{owner}/{repo}/issues/{index}/labels	Remove all labels from an issue
IssueApi	**issue_create_comment**	**POST** /repos/{owner}/{repo}/issues/{index}/comments	Add a comment to an issue
IssueApi	**issue_create_issue**	**POST** /repos/{owner}/{repo}/issues	Create an issue. If using deadline only the date will be taken into account, and time of day ignored.
IssueApi	**issue_create_label**	**POST** /repos/{owner}/{repo}/labels	Create a label
IssueApi	**issue_create_milestone**	**POST** /repos/{owner}/{repo}/milestones	Create a milestone
IssueApi	**issue_delete_comment**	**DELETE** /repos/{owner}/{repo}/issues/comments/{id}	Delete a comment
IssueApi	**issue_delete_comment_deprecated**	**DELETE** /repos/{owner}/{repo}/issues/{index}/comments/{id}	Delete a comment
IssueApi	**issue_delete_comment_reaction**	**DELETE** /repos/{owner}/{repo}/issues/comments/{id}/reactions	Remove a reaction from a comment of an issue
IssueApi	**issue_delete_issue_reaction**	**DELETE** /repos/{owner}/{repo}/issues/{index}/reactions	Remove a reaction from an issue
IssueApi	**issue_delete_label**	**DELETE** /repos/{owner}/{repo}/labels/{id}	Delete a label
IssueApi	**issue_delete_milestone**	**DELETE** /repos/{owner}/{repo}/milestones/{id}	Delete a milestone
IssueApi	**issue_delete_stop_watch**	**DELETE** /repos/{owner}/{repo}/issues/{index}/stopwatch/delete	Delete an issue's existing stopwatch.
IssueApi	**issue_delete_subscription**	**DELETE** /repos/{owner}/{repo}/issues/{index}/subscriptions/{user}	Unsubscribe user from issue
IssueApi	**issue_delete_time**	**DELETE** /repos/{owner}/{repo}/issues/{index}/times/{id}	Delete specific tracked time
IssueApi	**issue_edit_comment**	**PATCH** /repos/{owner}/{repo}/issues/comments/{id}	Edit a comment
IssueApi	**issue_edit_comment_deprecated**	**PATCH** /repos/{owner}/{repo}/issues/{index}/comments/{id}	Edit a comment
IssueApi	**issue_edit_issue**	**PATCH** /repos/{owner}/{repo}/issues/{index}	Edit an issue. If using deadline only the date will be taken into account, and time of day ignored.

Class	Method	HTTP request	Description
IssueApi	**issue_edit_issue_deadline**	**POST** /repos/{owner}/{repo}/issues/{index}/deadline	Set an issue deadline. If set to null, the deadline is deleted. If using deadline only the date will be taken into account, and time of day ignored.
IssueApi	**issue_edit_label**	**PATCH** /repos/{owner}/{repo}/labels/{id}	Update a label
IssueApi	**issue_edit_milestone**	**PATCH** /repos/{owner}/{repo}/milestones/{id}	Update a milestone
IssueApi	**issue_get_comment**	**GET** /repos/{owner}/{repo}/issues/comments/{id}	Get a comment
IssueApi	**issue_get_comment_reactions**	**GET** /repos/{owner}/{repo}/issues/comments/{id}/reactions	Get a list of reactions from a comment of an issue
IssueApi	**issue_get_comments**	**GET** /repos/{owner}/{repo}/issues/{index}/comments	List all comments on an issue
IssueApi	**issue_get_issue**	**GET** /repos/{owner}/{repo}/issues/{index}	Get an issue
IssueApi	**issue_get_issue_reactions**	**GET** /repos/{owner}/{repo}/issues/{index}/reactions	Get a list reactions of an issue
IssueApi	**issue_get_label**	**GET** /repos/{owner}/{repo}/labels/{id}	Get a single label
IssueApi	**issue_get_labels**	**GET** /repos/{owner}/{repo}/issues/{index}/labels	Get an issue's labels
IssueApi	**issue_get_milestone**	**GET** /repos/{owner}/{repo}/milestones/{id}	Get a milestone
IssueApi	**issue_get_milestones_list**	**GET** /repos/{owner}/{repo}/milestones	Get all of a repository's opened milestones
IssueApi	**issue_get_repo_comments**	**GET** /repos/{owner}/{repo}/issues/comments	List all comments in a repository
IssueApi	**issue_list_issues**	**GET** /repos/{owner}/{repo}/issues	List a repository's issues
IssueApi	**issue_list_labels**	**GET** /repos/{owner}/{repo}/labels	Get all of a repository's labels
IssueApi	**issue_post_comment_reaction**	**POST** /repos/{owner}/{repo}/issues/comments/{id}/reactions	Add a reaction to a comment of an issue
IssueApi	**issue_post_issue_reaction**	**POST** /repos/{owner}/{repo}/issues/{index}/reactions	Add a reaction to an issue
IssueApi	**issue_remove_label**	**DELETE** /repos/{owner}/{repo}/issues/{index}/labels/{id}	Remove a label from an issue
IssueApi	**issue_replace_labels**	**PUT** /repos/{owner}/{repo}/issues/{index}/labels	Replace an issue's labels
IssueApi	**issue_reset_time**	**DELETE** /repos/{owner}/{repo}/issues/{index}/times	Reset a tracked time of an issue
IssueApi	**issue_search_issues**	**GET** /repos/issues/search	Search for issues across the repositories that the user has access to

Class	Method	HTTP request	Description
IssueApi	**issue_start_stop_watch**	**POST** /repos/{owner}/{repo}/issues/{index}/stopwatch/start	Start stopwatch on an issue.
IssueApi	**issue_stop_stop_watch**	**POST** /repos/{owner}/{repo}/issues/{index}/stopwatch/stop	Stop an issue's existing stopwatch.
IssueApi	**issue_subscriptions**	**GET** /repos/{owner}/{repo}/issues/{index}/subscriptions	Get users who subscribed on an issue.
IssueApi	**issue_tracked_times**	**GET** /repos/{owner}/{repo}/issues/{index}/times	List an issue's tracked times
MiscellaneousApi	**get_signing_key**	**GET** /signing-key.gpg	Get default signing-key.gpg
MiscellaneousApi	**get_version**	**GET** /version	Returns the version of the Gitea application
MiscellaneousApi	**render_markdown**	**POST** /markdown	Render a markdown document as HTML
MiscellaneousApi	**render_markdown_raw**	**POST** /markdown/raw	Render raw markdown as HTML
NotificationApi	**notify_get_list**	**GET** /notifications	List users's notification threads
NotificationApi	**notify_get_repo_list**	**GET** /repos/{owner}/{repo}/notifications	List users's notification threads on a specific repo
NotificationApi	**notify_get_thread**	**GET** /notifications/threads/{id}	Get notification thread by ID
NotificationApi	**notify_new_available**	**GET** /notifications/new	Check if unread notifications exist
NotificationApi	**notify_read_list**	**PUT** /notifications	Mark notification threads as read, pinned or unread
NotificationApi	**notify_read_repo_list**	**PUT** /repos/{owner}/{repo}/notifications	Mark notification threads as read, pinned or unread on a specific repo
NotificationApi	**notify_read_thread**	**PATCH** /notifications/threads/{id}	Mark notification thread as read by ID
OrganizationApi	**create_org_repo**	**POST** /orgs/{org}/repos	Create a repository in an organization
OrganizationApi	**create_org_repo_deprecated**	**POST** /org/{org}/repos	Create a repository in an organization
OrganizationApi	**org_add_team_member**	**PUT** /teams/{id}/members/{username}	Add a team member
OrganizationApi	**org_add_team_repository**	**PUT** /teams/{id}/repos/{org}/{repo}	Add a repository to a team
OrganizationApi	**org_conceal_member**	**DELETE** /orgs/{org}/public_members/{username}	Conceal a user's membership

Class	Method	HTTP request	Description
OrganizationApi	**org_create**	**POST** /orgs	Create an organization
OrganizationApi	**org_create_hook**	**POST** /orgs/{org}/hooks/	Create a hook
OrganizationApi	**org_create_label**	**POST** /orgs/{org}/labels	Create a label for an organization
OrganizationApi	**org_create_team**	**POST** /orgs/{org}/teams	Create a team
OrganizationApi	**org_delete**	**DELETE** /orgs/{org}	Delete an organization
OrganizationApi	**org_delete_hook**	**DELETE** /orgs/{org}/hooks/{id}	Delete a hook
OrganizationApi	**org_delete_label**	**DELETE** /orgs/{org}/labels/{id}	Delete a label
OrganizationApi	**org_delete_member**	**DELETE** /orgs/{org}/members/{username}	Remove a member from an organization
OrganizationApi	**org_delete_team**	**DELETE** /teams/{id}	Delete a team
OrganizationApi	**org_edit**	**PATCH** /orgs/{org}	Edit an organization
OrganizationApi	**org_edit_hook**	**PATCH** /orgs/{org}/hooks/{id}	Update a hook
OrganizationApi	**org_edit_label**	**PATCH** /orgs/{org}/labels/{id}	Update a label
OrganizationApi	**org_edit_team**	**PATCH** /teams/{id}	Edit a team
OrganizationApi	**org_get**	**GET** /orgs/{org}	Get an organization
OrganizationApi	**org_get_all**	**GET** /orgs	Get list of organizations
OrganizationApi	**org_get_hook**	**GET** /orgs/{org}/hooks/{id}	Get a hook
OrganizationApi	**org_get_label**	**GET** /orgs/{org}/labels/{id}	Get a single label
OrganizationApi	**org_get_team**	**GET** /teams/{id}	Get a team
OrganizationApi	**org_is_member**	**GET** /orgs/{org}/members/{username}	Check if a user is a member of an organization
OrganizationApi	**org_is_public_member**	**GET** /orgs/{org}/public_members/{username}	Check if a user is a public member of an organization
OrganizationApi	**org_list_current_user_orgs**	**GET** /user/orgs	List the current user's organizations
OrganizationApi	**org_list_hooks**	**GET** /orgs/{org}/hooks	List an organization's webhooks
OrganizationApi	**org_list_labels**	**GET** /orgs/{org}/labels	List an organization's labels
OrganizationApi	**org_list_members**	**GET** /orgs/{org}/members	List an organization's members
OrganizationApi	**org_list_public_members**	**GET** /orgs/{org}/public_members	List an organization's public members

Class	Method	HTTP request	Description
OrganizationApi	**org_list_repos**	**GET** /orgs/{org}/repos	List an organization's repos
OrganizationApi	**org_list_team_member**	**GET** /teams/{id}/members/{username}	List a particular member of team
OrganizationApi	**org_list_team_members**	**GET** /teams/{id}/members	List a team's members
OrganizationApi	**org_list_team_repos**	**GET** /teams/{id}/repos	List a team's repos
OrganizationApi	**org_list_teams**	**GET** /orgs/{org}/teams	List an organization's teams
OrganizationApi	**org_list_user_orgs**	**GET** /users/{username}/orgs	List a user's organizations
OrganizationApi	**org_publicize_member**	**PUT** /orgs/{org}/public_members/{username}	Publicize a user's membership
OrganizationApi	**org_remove_team_member**	**DELETE** /teams/{id}/members/{username}	Remove a team member
OrganizationApi	**org_remove_team_repository**	**DELETE** /teams/{id}/repos/{org}/{repo}	Remove a repository from a team
OrganizationApi	**team_search**	**GET** /orgs/{org}/teams/search	Search for teams within an organization
RepositoryApi	**create_current_user_repo**	**POST** /user/repos	Create a repository
RepositoryApi	**create_fork**	**POST** /repos/{owner}/{repo}/forks	Fork a repository
RepositoryApi	**get_annotated_tag**	**GET** /repos/{owner}/{repo}/git/tags/{sha}	Gets the tag object of an annotated tag (not lightweight tags)
RepositoryApi	**get_blob**	**GET** /repos/{owner}/{repo}/git/blobs/{sha}	Gets the blob of a repository.
RepositoryApi	**get_tree**	**GET** /repos/{owner}/{repo}/git/trees/{sha}	Gets the tree of a repository.
RepositoryApi	**list_forks**	**GET** /repos/{owner}/{repo}/forks	List a repository's forks
RepositoryApi	**repo_add_collaborator**	**PUT** /repos/{owner}/{repo}/collaborators/{collaborator}	Add a collaborator to a repository
RepositoryApi	**repo_add_team**	**PUT** /repos/{owner}/{repo}/teams/{team}	Add a team to a repository
RepositoryApi	**repo_add_topc**	**PUT** /repos/{owner}/{repo}/topics/{topic}	Add a topic to a repository
RepositoryApi	**repo_check_collaborator**	**GET** /repos/{owner}/{repo}/collaborators/{collaborator}	Check if a user is a collaborator of a repository
RepositoryApi	**repo_check_team**	**GET** /repos/{owner}/{repo}/teams/{team}	Check if a team is assigned to a repository
RepositoryApi	**repo_create_branch**	**POST** /repos/{owner}/{repo}/branches	Create a branch

Class	Method	HTTP request	Description
RepositoryApi	**repo_create_branch_protection**	**POST** /repos/{owner}/{repo}/branch_protections	Create a branch protections for a repository
RepositoryApi	**repo_create_file**	**POST** /repos/{owner}/{repo}/contents/{filepath}	Create a file in a repository
RepositoryApi	**repo_create_hook**	**POST** /repos/{owner}/{repo}/hooks	Create a hook
RepositoryApi	**repo_create_key**	**POST** /repos/{owner}/{repo}/keys	Add a key to a repository
RepositoryApi	**repo_create_pull_request**	**POST** /repos/{owner}/{repo}/pulls	Create a pull request
RepositoryApi	**repo_create_pull_review**	**POST** /repos/{owner}/{repo}/pulls/{index}/reviews	Create a review to an pull request
RepositoryApi	**repo_create_pull_review_requests**	**POST** /repos/{owner}/{repo}/pulls/{index}/requested_reviewers	create review requests for a pull request
RepositoryApi	**repo_create_release**	**POST** /repos/{owner}/{repo}/releases	Create a release
RepositoryApi	**repo_create_release_attachment**	**POST** /repos/{owner}/{repo}/releases/{id}/assets	Create a release attachment
RepositoryApi	**repo_create_status**	**POST** /repos/{owner}/{repo}/statuses/{sha}	Create a commit status
RepositoryApi	**repo_create_tag**	**POST** /repos/{owner}/{repo}/tags	Create a new git tag in a repository
RepositoryApi	**repo_delete**	**DELETE** /repos/{owner}/{repo}	Delete a repository
RepositoryApi	**repo_delete_branch**	**DELETE** /repos/{owner}/{repo}/branches/{branch}	Delete a specific branch from a repository
RepositoryApi	**repo_delete_branch_protection**	**DELETE** /repos/{owner}/{repo}/branch_protections/{name}	Delete a specific branch protection for the repository
RepositoryApi	**repo_delete_collaborator**	**DELETE** /repos/{owner}/{repo}/collaborators/{collaborator}	Delete a collaborator from a repository
RepositoryApi	**repo_delete_file**	**DELETE** /repos/{owner}/{repo}/contents/{filepath}	Delete a file in a repository
RepositoryApi	**repo_delete_git_hook**	**DELETE** /repos/{owner}/{repo}/hooks/git/{id}	Delete a Git hook in a repository
RepositoryApi	**repo_delete_hook**	**DELETE** /repos/{owner}/{repo}/hooks/{id}	Delete a hook in a repository
RepositoryApi	**repo_delete_key**	**DELETE** /repos/{owner}/{repo}/keys/{id}	Delete a key from a repository
RepositoryApi	**repo_delete_pull_review**	**DELETE** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}	Delete a specific review from a pull request
RepositoryApi	**repo_delete_pull_review_requests**	**DELETE** /repos/{owner}/{repo}/pulls/{index}/requested_reviewers	cancel review requests for a pull request
RepositoryApi	**repo_delete_release**	**DELETE** /repos/{owner}/{repo}/releases/{id}	Delete a release
RepositoryApi	**repo_delete_release_attachment**	**DELETE** /repos/{owner}/{repo}/releases/{id}/assets/{attachment_id}	Delete a release attachment

Class	Method	HTTP request	Description
RepositoryApi	**repo_delete_release_by_tag**	**DELETE** /repos/{owner}/{repo}/releases/tags/{tag}	Delete a release by tag name
RepositoryApi	**repo_delete_tag**	**DELETE** /repos/{owner}/{repo}/tags/{tag}	Delete a repository's tag by name
RepositoryApi	**repo_delete_team**	**DELETE** /repos/{owner}/{repo}/teams/{team}	Delete a team from a repository
RepositoryApi	**repo_delete_topic**	**DELETE** /repos/{owner}/{repo}/topics/{topic}	Delete a topic from a repository
RepositoryApi	**repo_dismiss_pull_review**	**POST** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}/dismissals	Dismiss a review for a pull request
RepositoryApi	**repo_download_pull_diff**	**GET** /repos/{owner}/{repo}/pulls/{index}.diff	Get a pull request diff
RepositoryApi	**repo_download_pull_patch**	**GET** /repos/{owner}/{repo}/pulls/{index}.patch	Get a pull request patch file
RepositoryApi	**repo_edit**	**PATCH** /repos/{owner}/{repo}	Edit a repository's properties. Only fields that are set will be changed.
RepositoryApi	**repo_edit_branch_protection**	**PATCH** /repos/{owner}/{repo}/branch_protections/{name}	Edit a branch protections for a repository. Only fields that are set will be changed
RepositoryApi	**repo_edit_git_hook**	**PATCH** /repos/{owner}/{repo}/hooks/git/{id}	Edit a Git hook in a repository
RepositoryApi	**repo_edit_hook**	**PATCH** /repos/{owner}/{repo}/hooks/{id}	Edit a hook in a repository
RepositoryApi	**repo_edit_pull_request**	**PATCH** /repos/{owner}/{repo}/pulls/{index}	Update a pull request. If using deadline only the date will be taken into account, and time of day ignored.
RepositoryApi	**repo_edit_release**	**PATCH** /repos/{owner}/{repo}/releases/{id}	Update a release
RepositoryApi	**repo_edit_release_attachment**	**PATCH** /repos/{owner}/{repo}/releases/{id}/assets/{attachment_id}	Edit a release attachment
RepositoryApi	**repo_get**	**GET** /repos/{owner}/{repo}	Get a repository
RepositoryApi	**repo_get_all_commits**	**GET** /repos/{owner}/{repo}/commits	Get a list of all commits from a repository
RepositoryApi	**repo_get_archive**	**GET** /repos/{owner}/{repo}/archive/{archive}	Get an archive of a repository
RepositoryApi	**repo_get_assignees**	**GET** /repos/{owner}/{repo}/assignees	Return all users that have write access and can be assigned to issues

Class	Method	HTTP request	Description
RepositoryApi	**repo_get_branch**	**GET** /repos/{owner}/{repo}/branches/{branch}	Retrieve a specific branch from a repository, including its effective branch protection
RepositoryApi	**repo_get_branch_protection**	**GET** /repos/{owner}/{repo}/branch_protections/{name}	Get a specific branch protection for the repository
RepositoryApi	**repo_get_by_id**	**GET** /repositories/{id}	Get a repository by id
RepositoryApi	**repo_get_combined_status_by_ref**	**GET** /repos/{owner}/{repo}/commits/{ref}/status	Get a commit's combined status, by branch/tag/commit reference
RepositoryApi	**repo_get_contents**	**GET** /repos/{owner}/{repo}/contents/{filepath}	Gets the metadata and contents (if a file) of an entry in a repository, or a list of entries if a dir
RepositoryApi	**repo_get_contents_list**	**GET** /repos/{owner}/{repo}/contents	Gets the metadata of all the entries of the root dir
RepositoryApi	**repo_get_editor_config**	**GET** /repos/{owner}/{repo}/editorconfig/{filepath}	Get the EditorConfig definitions of a file in a repository
RepositoryApi	**repo_get_git_hook**	**GET** /repos/{owner}/{repo}/hooks/git/{id}	Get a Git hook
RepositoryApi	**repo_get_hook**	**GET** /repos/{owner}/{repo}/hooks/{id}	Get a hook
RepositoryApi	**repo_get_issue_templates**	**GET** /repos/{owner}/{repo}/issue_templates	Get available issue templates for a repository
RepositoryApi	**repo_get_key**	**GET** /repos/{owner}/{repo}/keys/{id}	Get a repository's key by id
RepositoryApi	**repo_get_languages**	**GET** /repos/{owner}/{repo}/languages	Get languages and number of bytes of code written
RepositoryApi	**repo_get_pull_request**	**GET** /repos/{owner}/{repo}/pulls/{index}	Get a pull request
RepositoryApi	**repo_get_pull_review**	**GET** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}	Get a specific review for a pull request
RepositoryApi	**repo_get_pull_review_comments**	**GET** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}/comments	Get a specific review for a pull request
RepositoryApi	**repo_get_raw_file**	**GET** /repos/{owner}/{repo}/raw/{filepath}	Get a file from a repository
RepositoryApi	**repo_get_release**	**GET** /repos/{owner}/{repo}/releases/{id}	Get a release
RepositoryApi	**repo_get_release_attachment**	**GET** /repos/{owner}/{repo}/releases/{id}/assets/{attachment_id}	Get a release attachment
RepositoryApi	**repo_get_release_by_tag**	**GET** /repos/{owner}/{repo}/releases/tags/{tag}	Get a release by tag name

Class	Method	HTTP request	Description
RepositoryApi	**repo_get_reviewers**	**GET** /repos/{owner}/{repo}/reviewers	Return all users that can be requested to review in this repo
RepositoryApi	**repo_get_single_commit**	**GET** /repos/{owner}/{repo}/git/commits/{sha}	Get a single commit from a repository
RepositoryApi	**repo_get_tag**	**GET** /repos/{owner}/{repo}/tags/{tag}	Get the tag of a repository by tag name
RepositoryApi	**repo_list_all_git_refs**	**GET** /repos/{owner}/{repo}/git/refs	Get specified ref or filtered repository's refs
RepositoryApi	**repo_list_branch_protection**	**GET** /repos/{owner}/{repo}/branch_protections	List branch protections for a repository
RepositoryApi	**repo_list_branches**	**GET** /repos/{owner}/{repo}/branches	List a repository's branches
RepositoryApi	**repo_list_collaborators**	**GET** /repos/{owner}/{repo}/collaborators	List a repository's collaborators
RepositoryApi	**repo_list_git_hooks**	**GET** /repos/{owner}/{repo}/hooks/git	List the Git hooks in a repository
RepositoryApi	**repo_list_git_refs**	**GET** /repos/{owner}/{repo}/git/refs/{ref}	Get specified ref or filtered repository's refs
RepositoryApi	**repo_list_hooks**	**GET** /repos/{owner}/{repo}/hooks	List the hooks in a repository
RepositoryApi	**repo_list_keys**	**GET** /repos/{owner}/{repo}/keys	List a repository's keys
RepositoryApi	**repo_list_pull_requests**	**GET** /repos/{owner}/{repo}/pulls	List a repo's pull requests
RepositoryApi	**repo_list_pull_reviews**	**GET** /repos/{owner}/{repo}/pulls/{index}/reviews	List all reviews for a pull request
RepositoryApi	**repo_list_release_attachments**	**GET** /repos/{owner}/{repo}/releases/{id}/assets	List release's attachments
RepositoryApi	**repo_list_releases**	**GET** /repos/{owner}/{repo}/releases	List a repo's releases
RepositoryApi	**repo_list_stargazers**	**GET** /repos/{owner}/{repo}/stargazers	List a repo's stargazers
RepositoryApi	**repo_list_statuses**	**GET** /repos/{owner}/{repo}/statuses/{sha}	Get a commit's statuses
RepositoryApi	**repo_list_statuses_by_ref**	**GET** /repos/{owner}/{repo}/commits/{ref}/statuses	Get a commit's statuses, by branch/tag/commit reference
RepositoryApi	**repo_list_subscribers**	**GET** /repos/{owner}/{repo}/subscribers	List a repo's watchers
RepositoryApi	**repo_list_tags**	**GET** /repos/{owner}/{repo}/tags	List a repository's tags
RepositoryApi	**repo_list_teams**	**GET** /repos/{owner}/{repo}/teams	List a repository's teams

Class	Method	HTTP request	Description
RepositoryApi	**repo_list_topics**	**GET** /repos/{owner}/{repo}/topics	Get list of topics that a repository has
RepositoryApi	**repo_merge_pull_request**	**POST** /repos/{owner}/{repo}/pulls/{index}/merge	Merge a pull request
RepositoryApi	**repo_migrate**	**POST** /repos/migrate	Migrate a remote git repository
RepositoryApi	**repo_mirror_sync**	**POST** /repos/{owner}/{repo}/mirror-sync	Sync a mirrored repository
RepositoryApi	**repo_pull_request_is_merged**	**GET** /repos/{owner}/{repo}/pulls/{index}/merge	Check if a pull request has been merged
RepositoryApi	**repo_search**	**GET** /repos/search	Search for repositories
RepositoryApi	**repo_signing_key**	**GET** /repos/{owner}/{repo}/signing-key.gpg	Get signing-key.gpg for given repository
RepositoryApi	**repo_submit_pull_review**	**POST** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}	Submit a pending review to an pull request
RepositoryApi	**repo_test_hook**	**POST** /repos/{owner}/{repo}/hooks/{id}/tests	Test a push webhook
RepositoryApi	**repo_tracked_times**	**GET** /repos/{owner}/{repo}/times	List a repo's tracked times
RepositoryApi	**repo_transfer**	**POST** /repos/{owner}/{repo}/transfer	Transfer a repo ownership
RepositoryApi	**repo_un_dismiss_pull_review**	**POST** /repos/{owner}/{repo}/pulls/{index}/reviews/{id}/undismissals	Cancel to dismiss a review for a pull request
RepositoryApi	**repo_update_file**	**PUT** /repos/{owner}/{repo}/contents/{filepath}	Update a file in a repository
RepositoryApi	**repo_update_pull_request**	**POST** /repos/{owner}/{repo}/pulls/{index}/update	Merge PR's baseBranch into headBranch
RepositoryApi	**repo_update_topics**	**PUT** /repos/{owner}/{repo}/topics	Replace list of topics for a repository
RepositoryApi	**topic_search**	**GET** /topics/search	search topics via keyword
RepositoryApi	**user_current_check_subscription**	**GET** /repos/{owner}/{repo}/subscription	Check if the current user is watching a repo
RepositoryApi	**user_current_delete_subscription**	**DELETE** /repos/{owner}/{repo}/subscription	Unwatch a repo
RepositoryApi	**user_current_put_subscription**	**PUT** /repos/{owner}/{repo}/subscription	Watch a repo
RepositoryApi	**user_tracked_times**	**GET** /repos/{owner}/{repo}/times/{user}	List a user's tracked times in a repo
SettingsApi	**get_general_api_settings**	**GET** /settings/api	Get instance's global settings for api

Class	Method	HTTP request	Description
SettingsApi	**get_general_attachment_settings**	**GET** /settings/attachment	Get instance's global settings for Attachment
SettingsApi	**get_general_repository_settings**	**GET** /settings/repository	Get instance's global settings for repositories
SettingsApi	**get_general_ui_settings**	**GET** /settings/ui	Get instance's global settings for ui
UserApi	**create_current_user_repo**	**POST** /user/repos	Create a repository
UserApi	**get_user_settings**	**GET** /user/settings	Get user settings
UserApi	**update_user_settings**	**PATCH** /user/settings	Update user settings
UserApi	**user_add_email**	**POST** /user/emails	Add email addresses
UserApi	**user_check_following**	**GET** /users/{follower}/following/{followee}	Check if one user is following another user
UserApi	**user_create_o_auth2_application**	**POST** /user/applications/oauth2	creates a new OAuth2 application
UserApi	**user_create_token**	**POST** /users/{username}/tokens	Create an access token
UserApi	**user_current_check_following**	**GET** /user/following/{username}	Check whether a user is followed by the authenticated user
UserApi	**user_current_check_starring**	**GET** /user/starred/{owner}/{repo}	Whether the authenticated is starring the repo
UserApi	**user_current_delete_follow**	**DELETE** /user/following/{username}	Unfollow a user
UserApi	**user_current_delete_gpg_key**	**DELETE** /user/gpg_keys/{id}	Remove a GPG key
UserApi	**user_current_delete_key**	**DELETE** /user/keys/{id}	Delete a public key
UserApi	**user_current_delete_star**	**DELETE** /user/starred/{owner}/{repo}	Unstar the given repo
UserApi	**user_current_get_gpg_key**	**GET** /user/gpg_keys/{id}	Get a GPG key
UserApi	**user_current_get_key**	**GET** /user/keys/{id}	Get a public key
UserApi	**user_current_list_followers**	**GET** /user/followers	List the authenticated user's followers
UserApi	**user_current_list_following**	**GET** /user/following	List the users that the authenticated user is following
UserApi	**user_current_list_gpg_keys**	**GET** /user/gpg_keys	List the authenticated user's GPG keys
UserApi	**user_current_list_keys**	**GET** /user/keys	List the authenticated user's public keys

Class	Method	HTTP request	Description
UserApi	**user_current_list_repos**	**GET** /user/repos	List the repos that the authenticated user owns or has access to
UserApi	**user_current_list_starred**	**GET** /user/starred	The repos that the authenticated user has starred
UserApi	**user_current_list_subscriptions**	**GET** /user/subscriptions	List repositories watched by the authenticated user
UserApi	**user_current_post_gpg_key**	**POST** /user/gpg_keys	Create a GPG key
UserApi	**user_current_post_key**	**POST** /user/keys	Create a public key
UserApi	**user_current_put_follow**	**PUT** /user/following/{username}	Follow a user
UserApi	**user_current_put_star**	**PUT** /user/starred/{owner}/{repo}	Star the given repo
UserApi	**user_current_tracked_times**	**GET** /user/times	List the current user's tracked times
UserApi	**user_delete_access_token**	**DELETE** /users/{username}/tokens/{token}	delete an access token
UserApi	**user_delete_email**	**DELETE** /user/emails	Delete email addresses
UserApi	**user_delete_o_auth2_application**	**DELETE** /user/applications/oauth2/{id}	delete an OAuth2 Application
UserApi	**user_get**	**GET** /users/{username}	Get a user
UserApi	**user_get_current**	**GET** /user	Get the authenticated user
UserApi	**user_get_heatmap_data**	**GET** /users/{username}/heatmap	Get a user's heatmap
UserApi	**user_get_o_auth2_application**	**GET** /user/applications/oauth2/{id}	get an OAuth2 Application
UserApi	**user_get_oauth2_application**	**GET** /user/applications/oauth2	List the authenticated user's oauth2 applications
UserApi	**user_get_stop_watches**	**GET** /user/stopwatches	Get list of all existing stopwatches
UserApi	**user_get_tokens**	**GET** /users/{username}/tokens	List the authenticated user's access tokens
UserApi	**user_list_emails**	**GET** /user/emails	List the authenticated user's email addresses
UserApi	**user_list_followers**	**GET** /users/{username}/followers	List the given user's followers
UserApi	**user_list_following**	**GET** /users/{username}/following	List the users that the given user is following

Class	Method	HTTP request	Description
UserApi	**user_list_gpg_keys**	**GET** /users/{username}/gpg_keys	List the given user's GPG keys
UserApi	**user_list_keys**	**GET** /users/{username}/keys	List the given user's public keys
UserApi	**user_list_repos**	**GET** /users/{username}/repos	List the repos owned by the given user
UserApi	**user_list_starred**	**GET** /users/{username}/starred	The repos that the given user has starred
UserApi	**user_list_subscriptions**	**GET** /users/{username}/subscriptions	List the repositories watched by a user
UserApi	**user_list_teams**	**GET** /user/teams	List all the teams a user belongs to
UserApi	**user_search**	**GET** /users/search	Search for users
UserApi	**user_update_o_auth2_application**	**PATCH** /user/applications/oauth2/{id}	update an OAuth2 Application, this includes regenerating the client secret

Documentation For Models

- APIError
- AccessToken
- AddCollaboratorOption
- AddTimeOption
- AnnotatedTag
- AnnotatedTagObject
- Attachment
- Branch
- BranchProtection
- CombinedStatus
- Comment
- Commit
- CommitAffectedFiles
- CommitDateOptions
- CommitMeta
- CommitStatus
- CommitStatusState
- CommitUser
- ContentsResponse
- CreateBranchProtectionOption
- CreateBranchRepoOption
- CreateEmailOption
- CreateFileOptions
- CreateForkOption
- CreateGPGKeyOption
- CreateHookOption
- CreateHookOptionConfig
- CreateIssueCommentOption
- CreateIssueOption
- CreateKeyOption
- CreateLabelOption
- CreateMilestoneOption
- CreateOAuth2ApplicationOptions

- CreateOrgOption
- CreatePullRequestOption
- CreatePullReviewComment
- CreatePullReviewOptions
- CreateReleaseOption
- CreateRepoOption
- CreateStatusOption
- CreateTagOption
- CreateTeamOption
- CreateUserOption
- Cron
- DeleteEmailOption
- DeleteFileOptions
- DeployKey
- DismissPullReviewOptions
- EditAttachmentOptions
- EditBranchProtectionOption
- EditDeadlineOption
- EditGitHookOption
- EditHookOption
- EditIssueCommentOption
- EditIssueOption
- EditLabelOption
- EditMilestoneOption
- EditOrgOption
- EditPullRequestOption
- EditReactionOption
- EditReleaseOption
- EditRepoOption
- EditTeamOption
- EditUserOption
- Email
- ExternalTracker
- ExternalWiki
- FileCommitResponse
- FileDeleteResponse
- FileLinksResponse
- FileResponse
- GPGKey
- GPGKeyEmail
- GeneralAPISettings
- GeneralAttachmentSettings
- GeneralRepoSettings
- GeneralUISettings
- GitBlobResponse
- GitEntry
- GitHook
- GitObject
- GitServiceType
- GitTreeResponse
- Hook
- Identity
- InternalTracker
- Issue
- IssueDeadline
- IssueLabelsOption
- IssueTemplate
- Label
- MarkdownOption
- MergePullRequestOption
- MigrateRepoForm
- MigrateRepoOptions
- Milestone

- NotificationCount
- NotificationSubject
- NotificationThread
- OAuth2Application
- Organization
- PRBranchInfo
- PayloadCommit
- PayloadCommitVerification
- PayloadUser
- Permission
- PublicKey
- PullRequest
- PullRequestMeta
- PullReview
- PullReviewComment
- PullReviewRequestOptions
- Reaction
- Reference
- Release
- RepoCommit
- RepoTopicOptions
- Repository
- RepositoryMeta
- ReviewStateType
- SearchResults
- ServerVersion
- StateType
- StopWatch
- SubmitPullReviewOptions
- Tag
- Team
- TimeStamp
- TopicName
- TopicResponse
- TrackedTime
- TransferRepoOption
- UpdateFileOptions
- User
- UserHeatmapData
- UserSettings
- UserSettingsOptions
- WatchInfo

Documentation For Authorization

AccessToken

- **Type**: API key
- **API key parameter name**: access_token
- **Location**: URL query string

AuthorizationHeaderToken

- **Type**: API key
- **API key parameter name**: Authorization
- **Location**: HTTP header

BasicAuth

- **Type**: HTTP basic authentication

SudoHeader

- **Type**: API key

- **API key parameter name**: Sudo
- **Location**: HTTP header

SudoParam

- **Type**: API key
- **API key parameter name**: sudo
- **Location**: URL query string

TOTPHeader

- **Type**: API key
- **API key parameter name**: X-GITEA-OTP
- **Location**: HTTP header

Token

- **Type**: API key
- **API key parameter name**: token
- **Location**: URL query string

Author