API methods

by section by oauth scope

account

/api/v1/me	oauth
/api/v1/me/blocked	oauth
/api/v1/me/friends	oauth
/api/v1/me/karma	oauth
/api/v1/me/prefs	oauth
/api/v1/me/trophies	oauth
/prefs/blocked	oauth
/prefs/friends	oauth
/prefs/messaging	oauth
/prefs/trusted	oauth
/prefs/*where*	oauth

captcha

/api/needs_captcha	oauth

collections

/api/v1/collections/add_post_to_collection	oauth
/api/v1/collections/collection	oauth
/api/v1/collections/create_collection	oauth
/api/v1/collections/delete_collection	oauth
/api/v1/collections/follow_collection	oauth
/api/v1/collections/remove_post_in_collection	oauth
/api/v1/collections/reorder_collection	oauth
/api/v1/collections/subreddit_collections	oauth
/api/v1/collections/update_collection_description	oauth
/api/v1/collections/update_collection_display_layout	oauth
/api/v1/collections/update_collection_title	oauth

emoji

/api/v1/*subreddit*/emoji.json	oauth
/api/v1/*subreddit*/emoji/*emoji_name*	oauth
/api/v1/*subreddit*/emoji_asset_upload_s3.json	oauth
/api/v1/*subreddit*/emoji_custom_size	oauth
/api/v1/*subreddit*/emojis/all	oauth

flair

/api/clearflairtemplates	oauth
/api/deleteflair	oauth
/api/deleteflairtemplate	oauth
/api/flair	oauth
/api/flair_template_order	oauth
/api/flairconfig	oauth
/api/flaircsv	oauth
/api/flairlist	oauth
/api/flairselector	oauth
/api/flairtemplate	oauth
/api/flairtemplate_v2	oauth
/api/link_flair	oauth
/api/link_flair_v2	oauth
/api/selectflair	oauth
/api/setflairenabled	oauth
/api/user_flair	oauth
/api/user_flair_v2	oauth

This is automatically-generated documentation for the reddit API.

Please take care to respect our API access rules.

overview

listings

Many endpoints on reddit use the same protocol for controlling pagination and filtering. These endpoints are called Listings and share five common parameters: `after` / `before`, `limit`, `count`, and `show`.

Listings do not use page numbers because their content changes so frequently. Instead, they allow you to view slices of the underlying data. Listing JSON responses contain `after` and `before` fields which are equivalent to the "next" and "prev" buttons on the site and in combination with `count` can be used to page through the listing.

The common parameters are as follows:

- `after` / `before` - only one should be specified. these indicate the fullname of an item in the listing to use as the anchor point of the slice.
- `limit` - the maximum number of items to return in this slice of the listing.
- `count` - the number of items already seen in this listing. on the html site, the builder uses this to determine when to give values for `before` and `after` in the response.
- `show` - optional parameter; if `all` is passed, filters such as "hide links that I have voted on" will be disabled.

To page through a listing, start by fetching the first page without specifying values for `after` and `count`. The response will contain an `after` value which you can pass in the next request. It is a good idea, but not required, to send an updated value for `count` which should be the number of items already fetched.

modhashes

A modhash is a token that the reddit API requires to help prevent CSRF. Modhashes can be obtained via the /api/me.json call or in response data of listing endpoints.

The preferred way to send a modhash is to include an `X-Modhash` custom HTTP header with your requests.

Modhashes are not required when authenticated with OAuth.

fullnames

A fullname is a combination of a thing's type (e.g. `Link`) and its unique ID which forms a compact encoding of a globally unique ID on reddit.

Fullnames start with the type prefix for the object's type, followed by the thing's unique ID in base 36. For example, `t3_15bfi0`.

type prefixes

`t1_`	Comment

reddit gold

links & comments

listings

live threads

private messages

`t2_`	Account
`t3_`	Link
`t4_`	Message
`t5_`	Subreddit
`t6_`	Award

response body encoding

For legacy reasons, all JSON response bodies currently have `<`, `>`, and `&` replaced with `&lt;`, `&gt;`, and `&amp;`, respectively. If you wish to opt out of this behaviour, add a `raw_json=1` parameter to your request.

account

GET **/api/v1/me** identity

Returns the identity of the user.

GET **/api/v1/me/karma** mysubreddits

Return a breakdown of subreddit karma.

GET **/api/v1/me/prefs** identity

Return the preference settings of the logged in user

`fields`	A comma-separated list of items from this set:
	`beta`
	`threaded_messages`
	`hide_downs`
	`private_feeds`
	`activity_relevant_ads`
	`show_stylesheets`
	`profile_opt_out`
	`bad_comment_autocollapse`
	`third_party_site_data_personalized_content`
	`show_link_flair`
	`hide_ups`
	`show_trending`
	`top_karma_subreddits`
	`country_code`
	`theme_selector`
	`monitor_mentions`
	`email_comment_reply`
	`newwindow`
	`email_new_user_welcome`
	`research`
	`ignore_suggested_sort`
	`show_presence`
	`email_upvote_comment`
	`email_digests`
	`whatsapp_comment_reply`
	`num_comments`
	`feed_recommendations_enabled`
	`clickgadget`
	`use_global_defaults`
	`label_nsfw`
	`domain_details`
	`email_messages`
	`live_orangereds`
	`highlight_controversial`
	`mark_messages_read`
	`no_profanity`
	`email_unsubscribe_all`
	`whatsapp_enabled`
	`lang`
	`in_redesign_beta`

misc

moderation

new modmail

modnote

multis

```
live_bar_recommendations_enabled
third_party_data_personalized_ads
email_chat_request
allow_clicktracking
hide_from_robots
show_gold_expiration
show_twitter
compress
store_visits
video_autoplay
email_upvote_post
email_username_mention
media_preview
email_user_new_follower
nightmode
enable_default_themes
geopopular
third_party_site_data_personalized_ads
survey_last_seen_time
threaded_modmail
enable_followers
min_comment_score
public_votes
show_location_based_recommendations
email_post_reply
collapse_read_messages
show_flair
send_crosspost_messages
search_include_over_18
hide_ads
third_party_personalized_ads
min_link_score
over_18
sms_notifications_enabled
numsites
media
legacy_search
email_private_message
send_welcome_messages
email_community_discovery
highlight_new_comments
default_comment_sort
accept_pms
```

PATCH **/api/v1/me/prefs** account

This endpoint expects JSON data of this format

```
{
  "accept_pms": one of (`everyone`,
`whitelisted`),
  "activity_relevant_ads": boolean value,
  "allow_clicktracking": boolean value,
  "bad_comment_autocollapse": one of (`off`,
`low`, `medium`, `high`),
  "beta": boolean value,
  "clickgadget": boolean value,
  "collapse_read_messages": boolean value,
  "compress": boolean value,
  "country_code": one of (`WF`, `JP`, `JM`,
`JO`, `WS`, `JE`, `GW`, `GU`, `GT`, `GS`,
`GR`, `GQ`, `GP`, `GY`, `GG`, `GF`, `GE`,
`GD`, `GB`, `GA`, `GN`, `GM`, `GL`, `GI`,
`GH`, `PR`, `PS`, `PW`, `PT`, `PY`, `PA`,
`PF`, `PG`, `PE`, `PK`, `PH`, `PN`, `PL`,
`PM`, `ZM`, `ZA`, `ZZ`, `ZW`, `ME`, `MD`,
`MG`, `MF`, `MA`, `MC`, `MM`, `ML`, `MO`,
`MN`, `MH`, `MK`, `MU`, `MT`, `MW`, `MV`,
`MQ`, `MP`, `MS`, `MR`, `MY`, `MX`, `MZ`,
`FR`, `FI`, `FJ`, `FK`, `FM`, `FO`, `CK`,
`CI`, `CH`, `CO`, `CN`, `CM`, `CL`, `CC`,
`CA`, `CG`, `CF`, `CD`, `CZ`, `CY`, `CX`,
`CR`, `CW`, `CV`, `CU`, `SZ`, `SY`, `SX`,
`SS`, `SR`, `SV`, `ST`, `SK`, `SJ`, `SI`,
`SH`, `SO`, `SN`, `SM`, `SL`, `SC`, `SB`,
`SA`, `SG`, `SE`, `SD`, `YE`, `YT`, `LB`,
`LC`, `LA`, `LK`, `LI`, `LV`, `LT`, `LU`,
```

search

subreddits

users

widgets

```
`LR`, `LS`, `LY`, `VA`, `VC`, `VE`, `VG`,
`IQ`, `VI`, `IS`, `IR`, `IT`, `VN`, `IM`,
`IL`, `IO`, `IN`, `IE`, `ID`, `BD`, `BE`,
`BF`, `BG`, `BA`, `BB`, `BL`, `BM`, `BN`,
`BO`, `BH`, `BI`, `BJ`, `BT`, `BV`, `BW`,
`BQ`, `BR`, `BS`, `BY`, `BZ`, `RU`, `RW`,
`RS`, `RE`, `RO`, `OM`, `HR`, `HT`, `HU`,
`HK`, `HN`, `HM`, `EH`, `EE`, `EG`, `EC`,
`ET`, `ES`, `ER`, `UY`, `UZ`, `US`, `UM`,
`UG`, `UA`, `VU`, `NI`, `NL`, `NO`, `NA`,
`NC`, `NE`, `NF`, `NG`, `NZ`, `NP`, `NR`,
`NU`, `XK`, `XZ`, `XX`, `KG`, `KE`, `KI`,
`KH`, `KN`, `KM`, `KR`, `KP`, `KW`, `KZ`,
`KY`, `DO`, `DM`, `DJ`, `DK`, `DE`, `DZ`,
`TZ`, `TV`, `TW`, `TT`, `TR`, `TN`, `TO`,
`TL`, `TM`, `TJ`, `TK`, `TH`, `TF`, `TG`,
`TD`, `TC`, `AE`, `AD`, `AG`, `AF`, `AI`,
`AM`, `AL`, `AO`, `AN`, `AQ`, `AS`, `AR`,
`AU`, `AT`, `AW`, `AX`, `AZ`, `QA`),
  "default_comment_sort": one of
(`confidence`, `top`, `new`, `controversial`,
`old`, `random`, `qa`, `live`),
  "domain_details": boolean value,
  "email_chat_request": boolean value,
  "email_comment_reply": boolean value,
  "email_community_discovery": boolean value,
  "email_digests": boolean value,
  "email_messages": boolean value,
  "email_new_user_welcome": boolean value,
  "email_post_reply": boolean value,
  "email_private_message": boolean value,
  "email_unsubscribe_all": boolean value,
  "email_upvote_comment": boolean value,
  "email_upvote_post": boolean value,
  "email_user_new_follower": boolean value,
  "email_username_mention": boolean value,
  "enable_default_themes": boolean value,
  "enable_followers": boolean value,
  "feed_recommendations_enabled": boolean
value,
  "g": one of (`GLOBAL`, `US`, `AR`, `AU`,
`BG`, `CA`, `CL`, `CO`, `HR`, `CZ`, `FI`,
`FR`, `DE`, `GR`, `HU`, `IS`, `IN`, `IE`,
`IT`, `JP`, `MY`, `MX`, `NZ`, `PH`, `PL`,
`PT`, `PR`, `RO`, `RS`, `SG`, `ES`, `SE`,
`TW`, `TH`, `TR`, `GB`, `US_WA`, `US_DE`,
`US_DC`, `US_WI`, `US_WV`, `US_HI`, `US_FL`,
`US_WY`, `US_NH`, `US_NJ`, `US_NM`, `US_TX`,
`US_LA`, `US_NC`, `US_ND`, `US_NE`, `US_TN`,
`US_NY`, `US_PA`, `US_CA`, `US_NV`, `US_VA`,
`US_CO`, `US_AK`, `US_AL`, `US_AR`, `US_VT`,
`US_IL`, `US_GA`, `US_IN`, `US_IA`, `US_OK`,
`US_AZ`, `US_ID`, `US_CT`, `US_ME`, `US_MD`,
`US_MA`, `US_OH`, `US_UT`, `US_MO`, `US_MN`,
`US_MI`, `US_RI`, `US_KS`, `US_MT`, `US_MS`,
`US_SC`, `US_KY`, `US_OR`, `US_SD`),
  "hide_ads": boolean value,
  "hide_downs": boolean value,
  "hide_from_robots": boolean value,
  "hide_ups": boolean value,
  "highlight_controversial": boolean value,
  "highlight_new_comments": boolean value,
  "ignore_suggested_sort": boolean value,
  "in_redesign_beta": boolean value,
  "label_nsfw": boolean value,
  "lang": a valid IETF language tag
(underscore separated),
  "legacy_search": boolean value,
  "live_bar_recommendations_enabled": boolean
value,
  "live_orangereds": boolean value,
  "mark_messages_read": boolean value,
  "media": one of (`on`, `off`, `subreddit`),
  "media_preview": one of (`on`, `off`,
`subreddit`),
  "min_comment_score": an integer between
-100 and 100,
  "min_link_score": an integer between -100
and 100,
  "monitor_mentions": boolean value,
  "newwindow": boolean value,
  "nightmode": boolean value,
  "no_profanity": boolean value,
  "num_comments": an integer between 1 and
500,
  "numsites": an integer between 1 and 100,
  "over_18": boolean value,
  "private_feeds": boolean value,
  "profile_opt_out": boolean value,
  "public_votes": boolean value,
  "research": boolean value,
  "search_include_over_18": boolean value,
  "send_crosspost_messages": boolean value,
  "send_welcome_messages": boolean value,
  "show_flair": boolean value,
  "show_gold_expiration": boolean value,
  "show_link_flair": boolean value,
  "show_location_based_recommendations":
boolean value,
  "show_presence": boolean value,
  "show_stylesheets": boolean value,
  "show_trending": boolean value,
  "show_twitter": boolean value,
  "sms_notifications_enabled": boolean value,
  "store_visits": boolean value,
  "survey_last_seen_time": an integer,
  "theme_selector": subreddit name,
  "third_party_data_personalized_ads":
boolean value,
  "third_party_personalized_ads": boolean
value,
  "third_party_site_data_personalized_ads":
boolean value,

"third_party_site_data_personalized_content":
boolean value,
  "threaded_messages": boolean value,
  "threaded_modmail": boolean value,
  "top_karma_subreddits": boolean value,
  "use_global_defaults": boolean value,
  "video_autoplay": boolean value,
  "whatsapp_comment_reply": boolean value,
  "whatsapp_enabled": boolean value,
}
```

GET **/api/v1/me/trophies** identity

Return a list of trophies for the current user.

GET **/prefs/***where* read rss support

→ **/prefs/friends**
→ **/prefs/blocked**
→ **/prefs/messaging**
→ **/prefs/trusted**
→ **/api/v1/me/friends**
→ **/api/v1/me/blocked**

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string all
sr_detail	(optional) expand subreddits

captcha

GET **/api/needs_captcha** any

> Check whether ReCAPTCHAs are needed for API methods

collections

POST **/api/v1/collections/add_post_to_collection**

modposts

Add a post to a collection

`collection_id`	the UUID of a collection
`link_fullname`	a fullname of a link
`uh / X-Modhash header`	a modhash

GET **/api/v1/collections/collection** read

Fetch a collection including all the links

`collection_id`	the UUID of a collection
`include_links`	boolean value

POST **/api/v1/collections/create_collection** modposts

Create a collection

`description`	a string no longer than 500 characters
`display_layout`	one of (`TIMELINE`, `GALLERY`)
`sr_fullname`	a fullname of a subreddit
`title`	title of the submission. up to 300 characters long
`uh / X-Modhash header`	a modhash

POST **/api/v1/collections/delete_collection** modposts

Delete a collection

`collection_id`	the UUID of a collection
`uh / X-Modhash header`	a modhash

POST **/api/v1/collections/follow_collection** subscribe

Follow or unfollow a collection

To follow, `follow` should be True. To unfollow, `follow` should be False. The user must have access to the subreddit to be able to follow a collection within it.

`collection_id`	the UUID of a collection
`follow`	boolean value
`uh / X-Modhash header`	a modhash

POST **/api/v1/collections/remove_post_in_collection**

modposts

Remove a post from a collection

`collection_id`	the UUID of a collection

`link_fullname`	a fullname of a link
`uh / X-Modhash header`	a modhash

POST /api/v1/collections/reorder_collection modposts

Reorder posts in a collection

`collection_id`	the UUID of a collection
`link_ids`	the list of comma seperated link_ids in the order to set them in
`uh / X-Modhash header`	a modhash

GET /api/v1/collections/subreddit_collections read

Fetch collections for the subreddit

`sr_fullname`	a fullname of a subreddit

POST /api/v1/collections/update_collection_description modposts

Update a collection's description

`collection_id`	the UUID of a collection
`description`	a string no longer than 500 characters
`uh / X-Modhash header`	a modhash

POST /api/v1/collections/update_collection_display_layout modposts

Update a collection's display layout

`collection_id`	the UUID of a collection
`display_layout`	one of (`TIMELINE`, `GALLERY`)
`uh / X-Modhash header`	a modhash

POST /api/v1/collections/update_collection_title modposts

Update a collection's title

`collection_id`	the UUID of a collection
`title`	title of the submission. up to 300 characters long
`uh / X-Modhash header`	a modhash

emoji

POST /api/v1/*subreddit*/emoji.json structuredstyles

Add an emoji to the DB by posting a message on emoji_upload_q. A job processor that listens on a queue, uses the s3_key provided in the request to locate the image in S3 Temp Bucket and moves it to the PERM bucket. It also adds it to the DB using name as the column and sr_fullname as the key and sends the status on the websocket URL that is provided as part of this response.

This endpoint should also be used to update custom subreddit emojis with new images. If only the permissions on an emoji require updating the POST_emoji_permissions endpoint should be requested, instead.

`mod_flair_only`	boolean value
`name`	Name of the emoji to be created. It can be alphanumeric without any special characters except '-' & '_' and cannot exceed 24 characters.
`post_flair_allowed`	boolean value
`s3_key`	S3 key of the uploaded image which can be obtained from the S3 url. This is of the form subreddit/hash_value.
`user_flair_allowed`	boolean value

DELETE **/api/v1/*subreddit*/emoji/*emoji_name*** structuredstyles

Delete a Subreddit emoji. Remove the emoji from Cassandra and purge the assets from S3 and the image resizing provider.

POST **/api/v1/*subreddit*/emoji_asset_upload_s3.json** # structuredstyles

Acquire and return an upload lease to s3 temp bucket. The return value of this function is a json object containing credentials for uploading assets to S3 bucket, S3 url for upload request and the key to use for uploading. Using this lease the client will upload the emoji image to S3 temp bucket (included as part of the S3 URL).

This lease is used by S3 to verify that the upload is authorized.

`filepath`	name and extension of the image file e.g. image1.png
`mimetype`	mime type of the image e.g. image/png

POST **/api/v1/*subreddit*/emoji_custom_size** structuredstyles

Set custom emoji size.

Omitting width or height will disable custom emoji sizing.

`height`	an integer between 1 and 40 (default: 0)
`width`	an integer between 1 and 40 (default: 0)

GET **/api/v1/*subreddit*/emojis/all** read

Get all emojis for a SR. The response inclueds snoomojis as well as emojis for the SR specified in the request.

The response has 2 keys: - snoomojis - SR emojis

flair

POST **[/r/*subreddit*]/api/clearflairtemplates** modflair

`api_type`	the string `json`
`flair_type`	one of (`USER_FLAIR`, `LINK_FLAIR`)
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/deleteflair** modflair

`api_type`	the string `json`
`name`	a user by name
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/deleteflairtemplate** modflair

`api_type`	the string `json`
`flair_template_id`	
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/flair** modflair

`api_type`	the string `json`
`css_class`	a valid subreddit image name
`link`	a fullname of a link
`name`	a user by name
`text`	a string no longer than 64 characters
`uh / X-Modhash header`	a modhash

PATCH **[/r/*subreddit*]/api/flair_template_order** modflair

Update the order of flair templates in the specified subreddit.

Order should contain every single flair id for that flair type; omitting any id will result in a loss of data.

`flair_type`	one of (`USER_FLAIR`, `LINK_FLAIR`)
`subreddit`	subreddit name
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/flairconfig** modflair

`api_type`	the string `json`
`flair_enabled`	boolean value
`flair_position`	one of (`left`, `right`)
`flair_self_assign_enabled`	boolean value
`link_flair_position`	one of (`,left,right`)
`link_flair_self_assign_enabled`	boolean value
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/flaircsv** modflair

Change the flair of multiple users in the same subreddit with a single API call.

Requires a string 'flair_csv' which has up to 100 lines of the form '`user`,`flairtext`,`cssclass`' (Lines beyond the 100th are ignored).

If both `cssclass` and `flairtext` are the empty string for a given `user`, instead clears that user's flair.

Returns an array of objects indicating if each flair setting was applied, or a reason for the failure.

`flair_csv`	comma-seperated flair information
`uh / X-Modhash header`	a modhash

GET **[/r/*subreddit*]/api/flairlist** modflair

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 1000)
`name`	a user by name
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

POST **[/r/*subreddit*]/api/flairselector** flair

Return information about a users's flair options.

If `link` is given, return link flair options for an existing link. If `is_newlink` is True, return link flairs options for a new link submission. Otherwise, return user flair options for this subreddit.

The logged in user's flair is also returned. Subreddit moderators may give a user by `name` to instead retrieve that user's flair.

`is_newlink`	boolean value
`link`	a fullname of a link
`name`	a user by name

POST **[/r/*subreddit*]/api/flairtemplate** modflair

`api_type`	the string `json`
`css_class`	a valid subreddit image name
`flair_template_id`	
`flair_type`	one of (USER_FLAIR, LINK_FLAIR)
`text`	a string no longer than 64 characters
`text_editable`	boolean value
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/flairtemplate_v2** modflair

Create or update a flair template.

This new endpoint is primarily used for the redesign.

`allowable_content`	one of (`all`, `emoji`, `text`)

api_type	the string json
background_color	a 6-digit rgb hex color, e.g. #AABBCC
css_class	a valid subreddit image name
flair_template_id	
flair_type	one of (USER_FLAIR, LINK_FLAIR)
max_emojis	an integer between 1 and 10 (default: 10)
mod_only	boolean value
override_css	
text	a string no longer than 64 characters
text_color	one of (light, dark)
text_editable	boolean value
uh / X-Modhash header	a modhash

GET [/r/*subreddit*]/api/link_flair flair

Return list of available link flair for the current subreddit.

Will not return flair if the user cannot set their own link flair and they are not a moderator that can set flair.

GET [/r/*subreddit*]/api/link_flair_v2 flair

Return list of available link flair for the current subreddit.

Will not return flair if the user cannot set their own link flair and they are not a moderator that can set flair.

POST [/r/*subreddit*]/api/selectflair flair

api_type	the string json
background_color	a 6-digit rgb hex color, e.g. #AABBCC
css_class	a valid subreddit image name
flair_template_id	
link	a fullname of a link
name	a user by name
return_rtson	[all|only|none]: "all" saves attributes and returns rtjson "only" only returns rtjson"none" only saves attributes
text	a string no longer than 64 characters
text_color	one of (light, dark)
uh / X-Modhash header	a modhash

POST [/r/*subreddit*]/api/setflairenabled flair

api_type	the string json
flair_enabled	boolean value
uh / X-Modhash header	a modhash

GET **[/r/*subreddit*]/api/user_flair** flair

Return list of available user flair for the current subreddit.

Will not return flair if flair is disabled on the subreddit, the user cannot set their own flair, or they are not a moderator that can set flair.

GET **[/r/*subreddit*]/api/user_flair_v2** flair

Return list of available user flair for the current subreddit.

If user is not a mod of the subreddit, this endpoint filters out mod_only templates.

reddit gold

POST **/api/v1/gold/gild/*fullname*** creddits

`fullname`	fullname of a thing

POST **/api/v1/gold/give/*username*** creddits

`months`	an integer between 1 and 36
`username`	A valid, existing reddit username

links & comments

POST **/api/comment** any

Submit a new comment or reply to a message.

`parent` is the fullname of the thing being replied to. Its value changes the kind of object created by this request:

- the fullname of a Link: a top-level comment in that Link's thread. (requires `submit` scope)
- the fullname of a Comment: a comment reply to that comment. (requires `submit` scope)
- the fullname of a Message: a message reply to that message. (requires `privatemessages` scope)

`text` should be the raw markdown body of the comment or message.

To start a new message thread, use /api/compose.

`api_type`	the string `json`
`recaptcha_token`	a string
`return_rtjson`	boolean value
`richtext_json`	JSON data
`text`	raw markdown text
`thing_id`	fullname of parent thing
`uh / X-Modhash header`	a modhash

POST **/api/del** edit

Delete a Link or Comment.

`id`	fullname of a thing created by the user

uh / X-Modhash header	a modhash

POST /api/editusertext edit

Edit the body text of a comment or self-post.

api_type	the string `json`
return_rtjson	boolean value
richtext_json	JSON data
text	raw markdown text
thing_id	fullname of a thing
uh / X-Modhash header	a modhash

POST /api/follow_post subscribe

Follow or unfollow a post.

To follow, `follow` should be True. To unfollow, `follow` should be False. The user must have access to the subreddit to be able to follow a post within it.

follow	boolean: True to follow or False to unfollow
fullname	fullname of a link
uh / X-Modhash header	a modhash

POST /api/hide report

Hide a link.

This removes it from the user's default view of subreddit listings.

See also: /api/unhide.

id	A comma-separated list of link fullnames
uh / X-Modhash header	a modhash

GET [/r/*subreddit*]/api/info read

Return a listing of things specified by their fullnames.

Only Links, Comments, and Subreddits are allowed.

id	A comma-separated list of thing fullnames
sr_name	comma-delimited list of subreddit names
url	a valid URL

POST /api/lock modposts

Lock a link or comment.

Prevents a post or new child comments from receiving new comments.

See also: /api/unlock.

id	fullname of a thing
uh / X-Modhash header	a modhash

POST **/api/marknsfw** modposts

Mark a link NSFW.

See also: /api/unmarknsfw.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

GET **/api/morechildren** read

Retrieve additional comments omitted from a base comment tree.

When a comment tree is rendered, the most relevant comments are selected for display first. Remaining comments are stubbed out with "MoreComments" links. This API call is used to retrieve the additional comments represented by those stubs, up to 100 at a time.

The two core parameters required are `link` and `children`. `link` is the fullname of the link whose comments are being fetched. `children` is a comma-delimited list of comment ID36s that need to be fetched.

If `id` is passed, it should be the ID of the MoreComments object this call is replacing. This is needed only for the HTML UI's purposes and is optional otherwise.

NOTE: you may only make one request at a time to this API endpoint. Higher concurrency will result in an error being returned.

If `limit_children` is True, only return the children requested.

`depth` is the maximum depth of subtrees in the thread.

`api_type`	the string `json`
`children`	
`depth`	(optional) an integer
`id`	(optional) id of the associated MoreChildren object
`limit_children`	boolean value
`link_id`	fullname of a link
`sort`	one of (`confidence`, `top`, `new`, `controversial`, `old`, `random`, `qa`, `live`)

POST **/api/report** report

Report a link, comment or message. Reporting a thing brings it to the attention of the subreddit's moderators. Reporting a message sends it to a system for admin review. For links and comments, the thing is implicitly hidden as well (see /api/hide for details).

See /r/{subreddit}/about/rules for for more about subreddit rules, and /r/{subreddit}/about for more about `free_form_reports`.

`additional_info`	a string no longer than 2000 characters
`api_type`	the string `json`
`custom_text`	a string no longer than 2000 characters
`from_help_desk`	boolean value
`from_modmail`	boolean value
`modmail_conv_id`	base36 modmail conversation id
`other_reason`	a string no longer than 100 characters

reason	a string no longer than 100 characters
rule_reason	a string no longer than 100 characters
site_reason	a string no longer than 100 characters
sr_name	a string no longer than 1000 characters
thing_id	fullname of a thing
uh / X-Modhash header	a modhash
usernames	A comma-separated list of items

POST /api/report_award report

award_id	a string
reason	a string no longer than 100 characters

POST /api/save save

Save a link or comment.

Saved things are kept in the user's saved listing for later perusal.

See also: /api/unsave.

category	a category name
id	fullname of a thing
uh / X-Modhash header	a modhash

GET /api/saved_categories save

Get a list of categories in which things are currently saved.

See also: /api/save.

POST /api/sendreplies edit

Enable or disable inbox replies for a link or comment.

`state` is a boolean that indicates whether you are enabling or disabling inbox replies - true to enable, false to disable.

id	fullname of a thing created by the user
state	boolean value
uh / X-Modhash header	a modhash

POST /api/set_contest_mode modposts

Set or unset "contest mode" for a link's comments.

`state` is a boolean that indicates whether you are enabling or disabling contest mode - true to enable, false to disable.

api_type	the string json
id	
state	boolean value
uh / X-Modhash header	a modhash

POST /api/set_subreddit_sticky modposts

Set or unset a Link as the sticky in its subreddit.

`state` is a boolean that indicates whether to sticky or unsticky this post - true to sticky, false to unsticky.

The `num` argument is optional, and only used when stickying a post. It allows specifying a particular "slot" to sticky the post into, and if there is already a post stickied in that slot it will be replaced. If there is no post in the specified slot to replace, or `num` is None, the bottom-most slot will be used.

api_type	the string json
id	
num	an integer between 1 and 4
state	boolean value
to_profile	boolean value
uh / X-Modhash header	a modhash

POST /api/set_suggested_sort modposts

Set a suggested sort for a link.

Suggested sorts are useful to display comments in a certain preferred way for posts. For example, casual conversation may be better sorted by new by default, or AMAs may be sorted by Q&A. A sort of an empty string clears the default sort.

api_type	the string json
id	
sort	one of (confidence, top, new, controversial, old, random, qa, live, blank)
uh / X-Modhash header	a modhash

POST /api/spoiler modposts

id	fullname of a link
uh / X-Modhash header	a modhash

POST /api/store_visits save

Requires a subscription to reddit premium

links	A comma-separated list of link fullnames
uh / X-Modhash header	a modhash

POST /api/submit submit

Submit a link to a subreddit.

Submit will create a link or self-post in the subreddit `sr` with the title `title`. If `kind` is `"link"`, then `url` is expected to be a valid URL to link to. Otherwise, `text`, if present, will be the body of the self-post unless `richtext_json` is present, in which case it will be converted into the body of the self-post. An error is thrown if both `text` and `richtext_json` are present.

`extension` is used for determining which view-type (e.g. `json`, `compact` etc.) to use for the redirect that is generated after submit.

`ad`	boolean value
`api_type`	the string `json`
`app`	
`collection_id`	(beta) the UUID of a collection
`extension`	extension used for redirects
`flair_id`	a string no longer than 36 characters
`flair_text`	a string no longer than 64 characters
`g-recaptcha-response`	
`kind`	one of (`link`, `self`, `image`, `video`, `videogif`)
`nsfw`	boolean value
`post_set_default_post_id`	a string
`post_set_id`	a string
`recaptcha_token`	a string
`resubmit`	boolean value
`richtext_json`	JSON data
`sendreplies`	boolean value
`spoiler`	boolean value
`sr`	subreddit name
`text`	raw markdown text
`title`	title of the submission. up to 300 characters long
`uh / X-Modhash header`	a modhash
`url`	a valid URL
`video_poster_url`	a valid URL

POST **/api/unhide** report

Unhide a link.

See also: /api/hide.

`id`	A comma-separated list of link fullnames
`uh / X-Modhash header`	a modhash

POST **/api/unlock** modposts

Unlock a link or comment.

Allow a post or comment to receive new comments.

See also: /api/lock.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/unmarknsfw** modposts

Remove the NSFW marking from a link.

See also: /api/marknsfw.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/unsave** save

Unsave a link or comment.

This removes the thing from the user's saved listings as well.

See also: /api/save.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/unspoiler** modposts

`id`	fullname of a link
`uh / X-Modhash header`	a modhash

POST **/api/vote** vote

Cast a vote on a thing.

`id` should be the fullname of the Link or Comment to vote on.

`dir` indicates the direction of the vote. Voting `1` is an upvote, `-1` is a downvote, and `0` is equivalent to "un-voting" by clicking again on a highlighted arrow.

Note: votes must be cast by humans. That is, API clients proxying a human's action one-for-one are OK, but bots deciding how to vote on content or amplifying a human's vote are not. See the reddit rules for more details on what constitutes vote cheating.

`dir`	vote direction. one of (1, 0, -1)
`id`	fullname of a thing
`rank`	an integer greater than 1
`uh / X-Modhash header`	a modhash

listings

GET **/best** read rss support

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

GET **/by_id/***names* read

Get a listing of links by fullname.

`names` is a list of fullnames for links separated by commas or spaces.

`names`	A comma-separated list of link fullnames

GET **[/r/***subreddit***]/comments/***article* read rss support

Get the comment tree for a given Link `article`.

If supplied, `comment` is the ID36 of a comment in the comment tree for `article`. This comment will be the (highlighted) focal point of the returned view and `context` will be the number of parents shown.

`depth` is the maximum depth of subtrees in the thread.

`limit` is the maximum number of comments to return.

See also: /api/morechildren and /api/comment.

`article`	ID36 of a link
`comment`	(optional) ID36 of a comment
`context`	an integer between 0 and 8
`depth`	(optional) an integer
`limit`	(optional) an integer
`showedits`	boolean value
`showmedia`	boolean value
`showmore`	boolean value
`showtitle`	boolean value
`sort`	one of (`confidence`, `top`, `new`, `controversial`, `old`, `random`, `qa`, `live`)
`sr_detail`	(optional) expand subreddits
`theme`	one of (`default`, `dark`)
`threaded`	boolean value
`truncate`	an integer between 0 and 50

GET **/duplicates/***article* read rss support

Return a list of other submissions of the same URL

This endpoint is a listing.

`after`	fullname of a thing
`article`	The base 36 ID of a Link
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`crossposts_only`	boolean value
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sort`	one of (`num_comments`, `new`)
`sr`	subreddit name

sr_detail	(optional) expand subreddits

GET [/r/*subreddit*]/hot read rss support

This endpoint is a listing.

g	one of (GLOBAL, US, AR, AU, BG, CA, CL, CO, HR, CZ, FI, FR, DE, GR, HU, IS, IN, IE, IT, JP, MY, MX, NZ, PH, PL, PT, PR, RO, RS, SG, ES, SE, TW, TH, TR, GB, US_WA, US_DE, US_DC, US_WI, US_WV, US_HI, US_FL, US_WY, US_NH, US_NJ, US_NM, US_TX, US_LA, US_NC, US_ND, US_NE, US_TN, US_NY, US_PA, US_CA, US_NV, US_VA, US_CO, US_AK, US_AL, US_AR, US_VT, US_IL, US_GA, US_IN, US_IA, US_OK, US_AZ, US_ID, US_CT, US_ME, US_MD, US_MA, US_OH, US_UT, US_MO, US_MN, US_MI, US_RI, US_KS, US_MT, US_MS, US_SC, US_KY, US_OR, US_SD)
after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string `all`
sr_detail	(optional) expand subreddits

GET [/r/*subreddit*]/new read rss support

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string `all`
sr_detail	(optional) expand subreddits

GET [/r/*subreddit*]/random read

The Serendipity button

GET [/r/*subreddit*]/rising read rss support

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string `all`
sr_detail	(optional) expand subreddits

GET [/r/*subreddit*]/*sort* read rss support

→ [/r/*subreddit*]/top
→ [/r/*subreddit*]/controversial

This endpoint is a listing.

t	one of (hour, day, week, month, year, all)
after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string all
sr_detail	(optional) expand subreddits

live threads

Real-time updates on reddit.

In addition to the standard reddit API, WebSockets play a huge role in reddit live. Receiving push notification of changes to the thread via websockets is much better than polling the thread repeatedly.

To connect to the websocket server, fetch /live/*thread*/about.json and get the `websocket_url` field. The websocket URL expires after a period of time; if it does, fetch a new one from that endpoint.

Once connected to the socket, a variety of messages can come in. All messages will be in text frames containing a JSON object with two keys: `type` and `payload`. Live threads can send messages with many `type`s:

- `update` - a new update has been posted in the thread. the `payload` contains the JSON representation of the update.
- `activity` - periodic update of the viewer counts for the thread.
- `settings` - the thread's settings have changed. the `payload` is an object with each key being a property of the thread (as in `about.json`) and its new value.
- `delete` - an update has been deleted (removed from the thread). the `payload` is the ID of the deleted update.
- `strike` - an update has been stricken (marked incorrect and crossed out). the `payload` is the ID of the stricken update.
- `embeds_ready` - a previously posted update has been parsed and embedded media is available for it now. the `payload` contains a `liveupdate_id` and list of `embeds` to add to it.
- `complete` - the thread has been marked complete. no further updates will be sent.

See /r/live for more information.

GET **/api/live/by_id/***names* read

Get a listing of live events by id.

names	a comma-delimited list of live thread fullnames or IDs

POST **/api/live/create** submit

Create a new live thread.

Once created, the initial settings can be modified with /api/live/*thread*/edit and new updates can be posted with /api/live/*thread*/update.

api_type	the string json

`description`	raw markdown text
`nsfw`	boolean value
`resources`	raw markdown text
`title`	a string no longer than 120 characters
`uh / X-Modhash header`	a modhash

GET **/api/live/happening_now** read

Get some basic information about the currently featured live thread.

Returns an empty 204 response for api requests if no thread is currently featured.

See also: /api/live/*thread*/about.

`show_announcements`	boolean value

POST **/api/live/*thread*/accept_contributor_invite** livemanage

Accept a pending invitation to contribute to the thread.

See also: /api/live/*thread*/leave_contributor.

`api_type`	the string `json`
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/close_thread** livemanage

Permanently close the thread, disallowing future updates.

Requires the `close` permission for this thread.

`api_type`	the string `json`
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/delete_update** edit

Delete an update from the thread.

Requires that specified update must have been authored by the user or that you have the `edit` permission for this thread.

See also: /api/live/*thread*/update.

`api_type`	the string `json`
`id`	the ID of a single update. e.g. `LiveUpdate_ff87068e-a126-11e3-9f93-12313b0b3603`
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/edit** livemanage

Configure the thread.

Requires the `settings` permission for this thread.

See also: /live/*thread*/about.json.

`api_type`	the string `json`
`description`	raw markdown text

`nsfw`	boolean value
`resources`	raw markdown text
`title`	a string no longer than 120 characters
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/hide_discussion** livemanage

Hide a linked comment thread from the discussions sidebar and listing.

Requires the `discussions` permission for this thread.

See also: /api/live/*thread*/unhide_discussion.

`api_type`	the string `json`
`link`	The base 36 ID of a Link
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/invite_contributor** livemanage

Invite another user to contribute to the thread.

Requires the `manage` permission for this thread. If the recipient accepts the invite, they will be granted the permissions specified.

See also: /api/live/*thread*/accept_contributor_invite, and /api/live/*thread*/rm_contributor_invite.

`api_type`	the string `json`
`name`	the name of an existing user
`permissions`	permission description e.g. `+update,+edit,-manage`
`type`	one of (`liveupdate_contributor_invite`, `liveupdate_contributor`)
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/leave_contributor** livemanage

Abdicate contributorship of the thread.

See also: /api/live/*thread*/accept_contributor_invite, and /api/live/*thread*/invite_contributor.

`api_type`	the string `json`
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/report** report

Report the thread for violating the rules of reddit.

`api_type`	the string `json`
`type`	one of (`spam`, `vote-manipulation`, `personal-information`, `sexualizing-minors`, `site-breaking`)
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/rm_contributor** livemanage

Revoke another user's contributorship.

Requires the `manage` permission for this thread.

See also: /api/live/*thread*/invite_contributor.

`api_type`	the string `json`
`id`	fullname of a account
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/rm_contributor_invite** livemanage

Revoke an outstanding contributor invite.

Requires the `manage` permission for this thread.

See also: /api/live/*thread*/invite_contributor.

`api_type`	the string `json`
`id`	fullname of a account
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/set_contributor_permissions**

livemanage

Change a contributor or contributor invite's permissions.

Requires the `manage` permission for this thread.

See also: /api/live/*thread*/invite_contributor and /api/live/*thread*/rm_contributor.

`api_type`	the string `json`
`name`	the name of an existing user
`permissions`	permission description e.g. `+update,+edit,-manage`
`type`	one of (`liveupdate_contributor_invite`, `liveupdate_contributor`)
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/strike_update** edit

Strike (mark incorrect and cross out) the content of an update.

Requires that specified update must have been authored by the user or that you have the `edit` permission for this thread.

See also: /api/live/*thread*/update.

`api_type`	the string `json`
`id`	the ID of a single update. e.g. `LiveUpdate_ff87068e-a126-11e3-9f93-12313b0b3603`
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/unhide_discussion** livemanage

Unhide a linked comment thread from the discussions sidebar and listing..

Requires the `discussions` permission for this thread.

See also: /api/live/*thread*/hide_discussion.

`api_type`	the string `json`
`link`	The base 36 ID of a Link
`uh / X-Modhash header`	a modhash

POST **/api/live/*thread*/update** submit

Post an update to the thread.

Requires the `update` permission for this thread.

See also: /api/live/*thread*/strike_update, and /api/live/*thread*/delete_update.

`api_type`	the string `json`
`body`	raw markdown text
`uh / X-Modhash header`	a modhash

GET **/live/*thread*** read rss support

Get a list of updates posted in this thread.

See also: /api/live/*thread*/update.

This endpoint is a listing.

`after`	the ID of a single update. e.g. `LiveUpdate_ff87068e-a126-11e3-9f93-12313b0b3603`
`before`	the ID of a single update. e.g. `LiveUpdate_ff87068e-a126-11e3-9f93-12313b0b3603`
`count`	a positive integer (default: 0)
`is_embed`	(internal use only)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`stylesr`	subreddit name

GET **/live/*thread*/about** read

Get some basic information about the live thread.

See also: /api/live/*thread*/edit.

GET **/live/*thread*/contributors** read

Get a list of users that contribute to this thread.

See also: /api/live/*thread*/invite_contributor, and /api/live/*thread*/rm_contributor.

GET **/live/*thread*/discussions** read rss support

Get a list of reddit submissions linking to this thread.

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)

show	(optional) the string `all`
sr_detail	(optional) expand subreddits

GET **/live/***thread***/updates/***update_id* read #

Get details about a specific update in a live thread.

private messages

POST **/api/block** privatemessages #

For blocking the author of a thing via inbox. Only accessible to approved OAuth applications

id	fullname of a thing
uh / X-Modhash header	a modhash

POST **/api/collapse_message** privatemessages #

Collapse a message

See also: /api/uncollapse_message

id	A comma-separated list of thing fullnames
uh / X-Modhash header	a modhash

POST **/api/compose** privatemessages #

Handles message composition under /message/compose.

api_type	the string `json`
from_sr	subreddit name
g-recaptcha-response	
subject	a string no longer than 100 characters
text	raw markdown text
to	the name of an existing user
uh / X-Modhash header	a modhash

POST **/api/del_msg** privatemessages #

Delete messages from the recipient's view of their inbox.

id	fullname of a thing
uh / X-Modhash header	a modhash

POST **/api/read_all_messages** privatemessages #

Queue up marking all messages for a user as read.

This may take some time, and returns 202 to acknowledge acceptance of the request.

filter_types	A comma-separated list of items
uh / X-Modhash	a modhash

header

POST /api/read_message privatemessages

`id`	A comma-separated list of thing fullnames
`uh / X-Modhash header`	a modhash

POST /api/unblock_subreddit privatemessages

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST /api/uncollapse_message privatemessages

Uncollapse a message

See also: /api/collapse_message

`id`	A comma-separated list of thing fullnames
`uh / X-Modhash header`	a modhash

POST /api/unread_message privatemessages

`id`	A comma-separated list of thing fullnames
`uh / X-Modhash header`	a modhash

GET /message/*where* privatemessages rss support

→ /message/inbox
→ /message/unread
→ /message/sent

This endpoint is a listing.

`mark`	one of (`true`, `false`)
`mid`	
`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

misc

GET [/r/*subreddit*]/api/saved_media_text submit

Retrieve the advisory text about saving media for relevant media links.

This endpoint returns a notice for display during the post submission process that is pertinent to media links.

`url`	a valid URL

GET **/api/v1/scopes** any

Retrieve descriptions of reddit's OAuth2 scopes.

If no scopes are given, information on all scopes are returned.

Invalid scope(s) will result in a 400 error with body that indicates the invalid scope(s).

`scopes`	(optional) An OAuth2 scope string

moderation

GET **[/r/*subreddit*]/about/log** modlog rss support

Get a list of recent moderation actions.

Moderator actions taken within a subreddit are logged. This listing is a view of that log with various filters to aid in analyzing the information.

The optional `mod` parameter can be a comma-delimited list of moderator names to restrict the results to, or the string `a` to restrict the results to admin actions taken within the subreddit.

The `type` parameter is optional and if sent limits the log entries returned to only those of the type specified.

This endpoint is a listing.

`after`	a ModAction ID
`before`	a ModAction ID
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 500)
`mod`	(optional) a moderator filter
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits
`type`	one of (`banuser`, `unbanuser`, `spamlink`, `removelink`, `approvelink`, `spamcomment`, `removecomment`, `approvecomment`, `addmoderator`, `showcomment`, `invitemoderator`, `uninvitemoderator`, `acceptmoderatorinvite`, `removemoderator`, `addcontributor`, `removecontributor`, `editsettings`, `editflair`, `distinguish`, `marknsfw`, `wikibanned`, `wikicontributor`, `wikiunbanned`, `wikipagelisted`, `removewikicontributor`, `wikirevise`, `wikipermlevel`, `ignorereports`, `unignorereports`, `setpermissions`, `setsuggestedsort`, `sticky`, `unsticky`, `setcontestmode`, `unsetcontestmode`, `lock`, `unlock`, `muteuser`, `unmuteuser`, `createrule`, `editrule`, `reorderrules`, `deleterule`, `spoiler`, `unspoiler`, `modmail_enrollment`, `community_status`, `community_styling`, `community_welcome_page`, `community_widgets`, `markoriginalcontent`, `collections`, `events`, `hidden_award`, `add_community_topics`, `remove_community_topics`, `create_scheduled_post`, `edit_scheduled_post`, `delete_scheduled_post`, `submit_scheduled_post`, `edit_comment_requirements`,

edit_post_requirements, invitesubscriber, submit_content_rating_survey, adjust_post_crowd_control_level, enable_post_crowd_control_filter, disable_post_crowd_control_filter, deleteoverriddenclassification, overrideclassification, reordermoderators, snoozereports, unsnoozereports, addnote, deletenote, addremovalreason, createremovalreason, updateremovalreason, deleteremovalreason, reorderremovalreason, dev_platform_app_changed, dev_platform_app_disabled, dev_platform_app_enabled, dev_platform_app_installed, dev_platform_app_uninstalled, edit_saved_response)

GET **[/r/*subreddit*]/about/*location*** read

→ **[/r/*subreddit*]/about/reports**
→ **[/r/*subreddit*]/about/spam**
→ **[/r/*subreddit*]/about/modqueue**
→ **[/r/*subreddit*]/about/unmoderated**
→ **[/r/*subreddit*]/about/edited**

Return a listing of posts relevant to moderators.

- reports: Things that have been reported.
- spam: Things that have been marked as spam or otherwise removed.
- modqueue: Things requiring moderator review, such as reported things and items caught by the spam filter.
- unmoderated: Things that have yet to be approved/removed by a mod.
- edited: Things that have been edited recently.

Requires the "posts" moderator permission for the subreddit.

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
location	
only	one of (links, comments, chat_comments)
show	(optional) the string all
sr_detail	(optional) expand subreddits

POST **[/r/*subreddit*]/api/accept_moderator_invite** modself

Accept an invite to moderate the specified subreddit.

The authenticated user must have been invited to moderate the subreddit by one of its current moderators.

See also: /api/friend and /subreddits/mine.

api_type	the string json
uh / X-Modhash header	a modhash

POST **/api/approve** modposts

Approve a link or comment.

If the thing was removed, it will be re-inserted into appropriate listings. Any reports on the approved thing will be discarded.

See also: /api/remove.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST /api/distinguish modposts

Distinguish a thing's author with a sigil.

This can be useful to draw attention to and confirm the identity of the user in the context of a link or comment of theirs. The options for distinguish are as follows:

- `yes` - add a moderator distinguish (`[M]`). only if the user is a moderator of the subreddit the thing is in.
- `no` - remove any distinguishes.
- `admin` - add an admin distinguish (`[A]`). admin accounts only.
- `special` - add a user-specific distinguish. depends on user.

The first time a top-level comment is moderator distinguished, the author of the link the comment is in reply to will get a notification in their inbox.

`sticky` is a boolean flag for comments, which will stick the distingushed comment to the top of all comments threads. If a comment is marked sticky, it will override any other stickied comment for that link (as only one comment may be stickied at a time.) Only top-level comments may be stickied.

`api_type`	the string `json`
`how`	one of (`yes`, `no`, `admin`, `special`)
`id`	fullname of a thing
`sticky`	boolean value
`uh / X-Modhash header`	a modhash

POST /api/ignore_reports modposts

Prevent future reports on a thing from causing notifications.

Any reports made about a thing after this flag is set on it will not cause notifications or make the thing show up in the various moderation listings.

See also: /api/unignore_reports.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST /api/leavecontributor modself

Abdicate approved user status in a subreddit.

See also: /api/friend.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/leavemoderator** modself

Abdicate moderator status in a subreddit.

See also: /api/friend.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/mute_message_author** modcontributors

For muting user via modmail.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/remove** modposts

Remove a link, comment, or modmail message.

If the thing is a link, it will be removed from all subreddit listings. If the thing is a comment, it will be redacted and removed from all subreddit comment listings.

See also: /api/approve.

`id`	fullname of a thing
`spam`	boolean value
`uh / X-Modhash header`	a modhash

POST **/api/show_comment** modposts

Mark a comment that it should not be collapsed because of crowd control.

The comment could still be collapsed for other reasons.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST **/api/snooze_reports** modposts

Prevent future reports on a thing from causing notifications.

For users who reported this thing (post, comment, etc) with the given report reason, reports from those users in the next 7 days will not be escalated to moderators. See also: /api/unsnooze_reports.

`id`	fullname of a thing
`reason`	
`uh / X-Modhash header`	a modhash

POST **/api/unignore_reports** modposts

Allow future reports on a thing to cause notifications.

See also: /api/ignore_reports.

`id`	fullname of a thing

`uh / X-Modhash header`	a modhash

POST /api/unmute_message_author modcontributors

For unmuting user via modmail.

`id`	fullname of a thing
`uh / X-Modhash header`	a modhash

POST /api/unsnooze_reports modposts

For users whose reports were snoozed (see /api/snooze_reports), to go back to escalating future reports from those users.

`id`	fullname of a thing
`reason`	
`uh / X-Modhash header`	a modhash

POST /api/update_crowd_control_level modposts

Change the post's crowd control level.

`id`	fullname of a thing
`level`	an integer between 0 and 3
`uh / X-Modhash header`	a modhash

GET [/r/*subreddit*]/stylesheet modconfig

Redirect to the subreddit's stylesheet if one exists.

See also: /api/subreddit_stylesheet.

new modmail

POST /api/mod/bulk_read modmail

Marks all conversations read for a particular conversation state within the passed list of subreddits.

`entity`	comma-delimited list of subreddit names
`state`	one of (`all`, `appeals`, `notifications`, `inbox`, `filtered`, `inprogress`, `mod`, `archived`, `default`, `highlighted`, `join_requests`, `new`)

GET /api/mod/conversations modmail

Get conversations for a logged in user or subreddits

`after`	A Modmail Conversation ID, in the form ModmailConversation_<id>
`entity`	comma-delimited list of subreddit names
`limit`	an integer between 1 and 100 (default: 25)
`sort`	one of (`recent`, `mod`, `user`, `unread`)
`state`	one of (`all`, `appeals`, `notifications`, `inbox`, `filtered`, `inprogress`, `mod`, `archived`, `default`, `highlighted`, `join_requests`, `new`)

POST **/api/mod/conversations** modmail

Creates a new conversation for a particular SR.

This endpoint will create a ModmailConversation object as well as the first ModmailMessage within the ModmailConversation object.

A note on `to`:

The `to` field for this endpoint is somewhat confusing. It can be:

- A User, passed like "username" or "u/username"
- A Subreddit, passed like "r/subreddit"
- null, meaning an internal moderator discussion

In this way `to` is a bit of a misnomer in modmail conversations. What it really means is the participant of the conversation who is not a mod of the subreddit.

`body`	raw markdown text
`isAuthorHidden`	boolean value
`srName`	subreddit name
`subject`	a string no longer than 100 characters
`to`	Modmail conversation recipient fullname

GET **/api/mod/conversations/:conversation_id** modmail

Returns all messages, mod actions and conversation metadata for a given conversation id

`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>
`markRead`	boolean value

POST **/api/mod/conversations/:conversation_id** modmail

Creates a new message for a particular conversation.

`body`	raw markdown text
`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>
`isAuthorHidden`	boolean value
`isInternal`	boolean value

POST **/api/mod/conversations/:conversation_id/approve** # modmail

Approve the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id

POST **/api/mod/conversations/:conversation_id/archive** # modmail

Marks a conversation as archived.

`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>

POST **/api/mod/conversations/:conversation_id/disapprove** modmail

Disapprove the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id

DELETE **/api/mod/conversations/:conversation_id/highlight**

modmail

Removes a highlight from a conversation.

`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>

POST **/api/mod/conversations/:conversation_id/highlight**

modmail

Marks a conversation as highlighted.

`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>

POST **/api/mod/conversations/:conversation_id/mute**

modmail

Mutes the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id
`num_hours`	one of (72, 168, 672)

POST **/api/mod/conversations/:conversation_id/temp_ban**

modmail

Temporary ban (switch from permanent to temporary ban) the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id
`duration`	an integer between 1 and 999

POST **/api/mod/conversations/:conversation_id/unarchive**

modmail

Marks conversation as unarchived.

`conversation_id`	A Modmail Conversation ID, in the form ModmailConversation_<id>

POST **/api/mod/conversations/:conversation_id/unban**

modmail

Unban the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id

POST **/api/mod/conversations/:conversation_id/unmute**

modmail

Unmutes the non mod user associated with a particular conversation.

`conversation_id`	base36 modmail conversation id

GET **/api/mod/conversations/:conversation_id/user**

modmail

Returns recent posts, comments and modmail conversations for a given user.

`conversation_id`	base36 modmail conversation id

POST **/api/mod/conversations/read** modmail

Marks a conversations as read for the user.

`conversationIds`	A comma-separated list of items

GET **/api/mod/conversations/subreddits** modmail

Returns a list of srs that the user moderates with mail permission

POST **/api/mod/conversations/unread** modmail

Marks conversations as unread for the user.

`conversationIds`	A comma-separated list of items

GET **/api/mod/conversations/unread/count** modmail

Endpoint to retrieve the unread conversation count by conversation state.

modnote

DELETE **/api/mod/notes** modnote

Delete a mod user note where type=NOTE.

Parameters should be passed as query parameters.

`note_id`	a unique ID for the note to be deleted (should have a ModNote_ prefix)
`subreddit`	subreddit name
`user`	account username

GET **/api/mod/notes** modnote

Get mod notes for a specific user in a given subreddit.

`before`	(optional) an encoded string used for pagination with mod notes
`filter`	(optional) one of (NOTE, APPROVAL, REMOVAL, BAN, MUTE, INVITE, SPAM, CONTENT_CHANGE, MOD_ACTION, ALL), to be used for querying specific types of mod notes (default: all)
`limit`	(optional) the number of mod notes to return in the response payload (default: 25, max: 100)'}
`subreddit`	subreddit name
`user`	account username

POST **/api/mod/notes** modnote

Create a mod user note where type=NOTE.

`label`	(optional) one of (BOT_BAN, PERMA_BAN, BAN, ABUSE_WARNING, SPAM_WARNING, SPAM_WATCH, SOLID_CONTRIBUTOR, HELPFUL_USER)
`note`	Content of the note, should be a string with a maximum character limit of 250
`reddit_id`	(optional) a fullname of a comment or post (should have either a t1 or t3 prefix)
`subreddit`	subreddit name
`user`	account username

GET **/api/mod/notes/recent** modnote

Fetch the most recent notes written by a moderator

Both parameters should be comma separated lists of equal lengths. The first subreddit will be paired with the first account to represent a query for a mod written note for that account in that subreddit and so forth for all subsequent pairs of subreddits and accounts. This request accepts up to 500 pairs of subreddit names and usernames. Parameters should be passed as query parameters.

The response will be a list of mod notes in the order that subreddits and accounts were given. If no note exist for a given subreddit/account pair, then null will take its place in the list.

`subreddits`	a comma delimited list of subreddits by name
`users`	a comma delimited list of usernames

multis

POST **/api/multi/copy**

subscribe #

Copy a multi.

Responds with 409 Conflict if the target already exists.

A "copied from ..." line will automatically be appended to the description.

`description_md`	raw markdown text
`display_name`	a string no longer than 50 characters
`expand_srs`	boolean value
`from`	multireddit url path
`to`	destination multireddit url path
`uh / X-Modhash header`	a modhash

GET **/api/multi/mine**

read #

Fetch a list of multis belonging to the current user.

`expand_srs`	boolean value

GET **/api/multi/user/***username*

read #

Fetch a list of public multis belonging to `username`

`expand_srs`	boolean value
`username`	A valid, existing reddit username

DELETE **/api/multi/***multipath*

→ **/api/filter/***filterpath*

subscribe #

Delete a multi.

`multipath`	multireddit url path
`uh / X-Modhash header`	a modhash
`expand_srs`	boolean value

GET **/api/multi/***multipath*

→ **/api/filter/***filterpath*

read #

Fetch a multi's data and subreddit list by name.

`expand_srs`	boolean value

`multipath`	multireddit url path

POST **/api/multi/***multipath* subscribe

→ **/api/filter/***filterpath*

Create a multi. Responds with 409 Conflict if it already exists.

`model` json data:

```
{
  "description_md": raw markdown text,
  "display_name": a string no longer than
50 characters,
  "icon_img": one of (`png`, `jpg`,
`jpeg`),
  "key_color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
  "subreddits": [
    {
      "name": subreddit name,
    },
    ...
  ],
  "visibility": one of (`private`,
`public`, `hidden`),
}
```

`multipath`	multireddit url path
`uh / X-Modhash header`	a modhash
`expand_srs`	boolean value

PUT **/api/multi/***multipath* subscribe

→ **/api/filter/***filterpath*

Create or update a multi.

`expand_srs`	boolean value

`model` json data:

```
{
  "description_md": raw markdown text,
  "display_name": a string no longer than
50 characters,
  "icon_img": one of (`png`, `jpg`,
`jpeg`),
  "key_color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
  "subreddits": [
    {
      "name": subreddit name,
    },
    ...
  ],
  "visibility": one of (`private`,
`public`, `hidden`),
}
```

`multipath`	multireddit url path
`uh / X-Modhash header`	a modhash

GET **/api/multi/***multipath***/description** read

Get a multi's description.

`multipath`	multireddit url path

PUT **/api/multi/***multipath***/description** read

Change a multi's markdown description.

`model`	json data:
	`{`
	`  "body_md": raw markdown text,`
	`}`

`multipath`	multireddit url path
`uh / X-Modhash header`	a modhash

DELETE **/api/multi/***multipath***/r/***srname* subscribe

→ **/api/filter/***filterpath***/r/***srname*

Remove a subreddit from a multi.

`multipath`	multireddit url path
`srname`	subreddit name
`uh / X-Modhash header`	a modhash

GET **/api/multi/***multipath***/r/***srname* read

→ **/api/filter/***filterpath***/r/***srname*

Get data about a subreddit in a multi.

`multipath`	multireddit url path
`srname`	subreddit name

PUT **/api/multi/***multipath***/r/***srname* subscribe

→ **/api/filter/***filterpath***/r/***srname*

Add a subreddit to a multi.

`model`	json data:
	`{`
	`  "name": subreddit name,`
	`}`

`multipath`	multireddit url path
`srname`	subreddit name
`uh / X-Modhash header`	a modhash

search

GET **[/r/***subreddit***]/search** read rss support

Search links page.

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`category`	a string no longer than 5 characters
`count`	a positive integer (default: 0)
`include_facets`	boolean value
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`q`	a string no longer than 512 characters

restrict_sr	boolean value
show	(optional) the string all
sort	one of (relevance, hot, top, new, comments)
sr_detail	(optional) expand subreddits
t	one of (hour, day, week, month, year, all)
type	(optional) comma-delimited list of result types (sr, link, user)

subreddits

GET **[/r/*subreddit*]/about/*where*** read rss support

→ **[/r/*subreddit*]/about/banned**
→ **[/r/*subreddit*]/about/muted**
→ **[/r/*subreddit*]/about/wikibanned**
→ **[/r/*subreddit*]/about/contributors**
→ **[/r/*subreddit*]/about/wikicontributors**
→ **[/r/*subreddit*]/about/moderators**

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string all
sr_detail	(optional) expand subreddits
user	A valid, existing reddit username

POST **[/r/*subreddit*]/api/delete_sr_banner** modconfig

Remove the subreddit's custom mobile banner.

See also: /api/upload_sr_img.

api_type	the string json
uh / X-Modhash header	a modhash

POST **[/r/*subreddit*]/api/delete_sr_header** modconfig

Remove the subreddit's custom header image.

The sitewide-default header image will be shown again after this call.

See also: /api/upload_sr_img.

api_type	the string json
uh / X-Modhash header	a modhash

POST **[/r/*subreddit*]/api/delete_sr_icon** modconfig

Remove the subreddit's custom mobile icon.

See also: /api/upload_sr_img.

api_type	the string json

`uh / X-Modhash header`	a modhash

POST [/r/*subreddit*]/api/delete_sr_img modconfig

Remove an image from the subreddit's custom image set.

The image will no longer count against the subreddit's image limit. However, the actual image data may still be accessible for an unspecified amount of time. If the image is currently referenced by the subreddit's stylesheet, that stylesheet will no longer validate and won't be editable until the image reference is removed.

See also: /api/upload_sr_img.

`api_type`	the string `json`
`img_name`	a valid subreddit image name
`uh / X-Modhash header`	a modhash

GET /api/recommend/sr/*srnames* read

DEPRECATED: Return subreddits recommended for the given subreddit(s).

Gets a list of subreddits recommended for `srnames`, filtering out any that appear in the optional `omit` param.

`omit`	comma-delimited list of subreddit names
`over_18`	boolean value
`srnames`	comma-delimited list of subreddit names

GET /api/search_reddit_names read

List subreddit names that begin with a query string.

Subreddits whose names begin with `query` will be returned. If `include_over_18` is false, subreddits with over-18 content restrictions will be filtered from the results.

If `include_unadvertisable` is False, subreddits that have `hide_ads` set to True or are on the `anti_ads_subreddits` list will be filtered.

If `exact` is true, only an exact match will be returned. Exact matches are inclusive of `over_18` subreddits, but not `hide_ad` subreddits when `include_unadvertisable` is `False`.

`exact`	boolean value
`include_over_18`	boolean value
`include_unadvertisable`	boolean value
`query`	a string up to 50 characters long, consisting of printable characters.
`search_query_id`	a uuid
`typeahead_active`	boolean value or None

POST /api/search_reddit_names read

List subreddit names that begin with a query string.

Subreddits whose names begin with `query` will be returned. If `include_over_18` is false, subreddits with over-18 content restrictions will be filtered from the results.

If `include_unadvertisable` is False, subreddits that have `hide_ads` set to True or are on the `anti_ads_subreddits` list will be filtered.

If `exact` is true, only an exact match will be returned. Exact matches are inclusive of `over_18` subreddits, but not `hide_ad` subreddits when `include_unadvertisable` is `False`.

`exact`	boolean value
`include_over_18`	boolean value
`include_unadvertisable`	boolean value
`query`	a string up to 50 characters long, consisting of printable characters.
`search_query_id`	a uuid
`typeahead_active`	boolean value or None

POST **/api/search_subreddits** read

List subreddits that begin with a query string.

Subreddits whose names begin with `query` will be returned. If `include_over_18` is false, subreddits with over-18 content restrictions will be filtered from the results.

If `include_unadvertisable` is False, subreddits that have `hide_ads` set to True or are on the `anti_ads_subreddits` list will be filtered.

If `exact` is true, only an exact match will be returned. Exact matches are inclusive of `over_18` subreddits, but not `hide_ad` subreddits when `include_unadvertisable` is `False`.

`exact`	boolean value
`include_over_18`	boolean value
`include_unadvertisable`	boolean value
`query`	a string up to 50 characters long, consisting of printable characters.
`search_query_id`	a uuid
`typeahead_active`	boolean value or None

POST **/api/site_admin** modconfig

Create or configure a subreddit.

If `sr` is specified, the request will attempt to modify the specified subreddit. If not, a subreddit with name `name` will be created.

This endpoint expects *all* values to be supplied on every request. If modifying a subset of options, it may be useful to get the current settings from /about/edit.json first.

For backwards compatibility, `description` is the sidebar text and `public_description` is the publicly visible subreddit description.

Most of the parameters for this endpoint are identical to options visible in the user interface and their meanings are best explained there.

See also: /about/edit.json.

`accept_followers`	boolean value
`admin_override_spam_comments`	boolean value
`admin_override_spam_links`	boolean value

admin_override_spam_selfposts	boolean value
all_original_content	boolean value
allow_chat_post_creation	boolean value
allow_discovery	boolean value
allow_galleries	boolean value
allow_images	boolean value
allow_polls	boolean value
allow_post_crossposts	boolean value
allow_prediction_contributors	boolean value
allow_predictions	boolean value
allow_predictions_tournament	boolean value
allow_talks	boolean value
allow_top	boolean value
allow_videos	boolean value
api_type	the string json
collapse_deleted_comments	boolean value

comment_contribution_settings	json data:
	{
	  "allowed_media_types": [
	    one of (`giphy`, `unknown`, `animated`, `static`, `expression`),
	    ...
	  ],
	}

comment_score_hide_mins	an integer between 0 and 1440 (default: 0)
crowd_control_chat_level	an integer between 0 and 3
crowd_control_filter	boolean value
crowd_control_level	an integer between 0 and 3
crowd_control_mode	boolean value
crowd_control_post_level	an integer between 0 and 3
description	raw markdown text
disable_contributor_requests	boolean value
exclude_banned_modqueue	boolean value
free_form_reports	boolean value
g-recaptcha-response	
hateful_content_threshold_abuse	an integer between 0 and 3
hateful_content_threshold_identity	an integer between 0 and 3
header-title	a string no longer than 500 characters
hide_ads	boolean value
key_color	a 6-digit rgb hex color, e.g. #AABBCC

link_type	one of (any, link, self)
modmail_harassment_filter_enabled	boolean value
name	subreddit name
new_pinned_post_pns_enabled	boolean value
original_content_tag_enabled	boolean value
over_18	boolean value
prediction_leaderboard_entry_type	an integer between 0 and 2
public_description	raw markdown text
restrict_commenting	boolean value
restrict_posting	boolean value
should_archive_posts	boolean value
show_media	boolean value
show_media_preview	boolean value
spam_comments	one of (low, high, all)
spam_links	one of (low, high, all)
spam_selfposts	one of (low, high, all)
spoilers_enabled	boolean value
sr	fullname of a thing
submit_link_label	a string no longer than 60 characters
submit_text	raw markdown text
submit_text_label	a string no longer than 60 characters

subreddit_discovery_settings	json data:
	{
	"disabled_discovery_types": [
	one of (`unknown`, `onboarding`),
	...
	],
	}

suggested_comment_sort	one of (confidence, top, new, controversial, old, random, qa, live)
title	a string no longer than 100 characters
toxicity_threshold_chat_level	an integer between 0 and 1
type	one of (gold_restricted, archived, restricted, private, employees_only, gold_only, public, user)
uh / X-Modhash header	a modhash
user_flair_pns_enabled	boolean value
welcome_message_enabled	boolean value
welcome_message_text	raw markdown text

`wiki_edit_age`	an integer between 0 and 36600 (default: 0)
`wiki_edit_karma`	an integer between 0 and 1000000000 (default: 0)
`wikimode`	one of (`disabled`, `modonly`, `anyone`)

GET **[/r/*subreddit*]/api/submit_text** submit

Get the submission text for the subreddit.

This text is set by the subreddit moderators and intended to be displayed on the submission form.

See also: /api/site_admin.

GET **/api/subreddit_autocomplete** read

Return a list of subreddits and data for subreddits whose names start with 'query'.

Uses typeahead endpoint to recieve the list of subreddits names. Typeahead provides exact matches, typo correction, fuzzy matching and boosts subreddits to the top that the user is subscribed to.

`include_over_18`	boolean value
`include_profiles`	boolean value
`query`	a string up to 25 characters long, consisting of printable characters.

GET **/api/subreddit_autocomplete_v2** read

`include_over_18`	boolean value
`include_profiles`	boolean value
`limit`	an integer between 1 and 10 (default: 5)
`query`	a string up to 25 characters long, consisting of printable characters.
`search_query_id`	a uuid
`typeahead_active`	boolean value or None

POST **[/r/*subreddit*]/api/subreddit_stylesheet** modconfig

Update a subreddit's stylesheet.

`op` should be `save` to update the contents of the stylesheet.

`api_type`	the string `json`
`op`	one of (`save`, `preview`)
`reason`	a string up to 256 characters long, consisting of printable characters.
`stylesheet_contents`	the new stylesheet content
`uh / X-Modhash header`	a modhash

POST **/api/subscribe** subscribe

Subscribe to or unsubscribe from a subreddit.

To subscribe, `action` should be `sub`. To unsubscribe, `action` should be `unsub`. The user must have access to the subreddit to be able to subscribe to it.

The `skip_initial_defaults` param can be set to True to prevent automatically subscribing the user to the current set of defaults when they take their first subscription action. Attempting to set it for an unsubscribe action will result in an error.

See also: /subreddits/mine/.

`action`	one of (`sub`, `unsub`)
`action_source`	one of (`onboarding`, `autosubscribe`)
`skip_initial_defaults`	boolean value
`sr / sr_name`	A comma-separated list of subreddit fullnames (when using the "sr" parameter), or of subreddit names (when using the "sr_name" parameter).
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/upload_sr_img** modconfig

Add or replace a subreddit image, custom header logo, custom mobile icon, or custom mobile banner.

- If the `upload_type` value is `img`, an image for use in the subreddit stylesheet is uploaded with the name specified in `name`.
- If the `upload_type` value is `header` then the image uploaded will be the subreddit's new logo and `name` will be ignored.
- If the `upload_type` value is `icon` then the image uploaded will be the subreddit's new mobile icon and `name` will be ignored.
- If the `upload_type` value is `banner` then the image uploaded will be the subreddit's new mobile banner and `name` will be ignored.

For backwards compatibility, if `upload_type` is not specified, the `header` field will be used instead:

- If the `header` field has value `0`, then `upload_type` is `img`.
- If the `header` field has value `1`, then `upload_type` is `header`.

The `img_type` field specifies whether to store the uploaded image as a PNG or JPEG.

Subreddits have a limited number of images that can be in use at any given time. If no image with the specified name already exists, one of the slots will be consumed.

If an image with the specified name already exists, it will be replaced. This does not affect the stylesheet immediately, but will take effect the next time the stylesheet is saved.

See also: /api/delete_sr_img, /api/delete_sr_header, /api/delete_sr_icon, and /api/delete_sr_banner.

`file`	file upload with maximum size of 500 KiB
`formid`	(optional) can be ignored
`header`	an integer between 0 and 1
`img_type`	one of `png` or `jpg` (default: `png`)
`name`	a valid subreddit image name
`uh / X-Modhash header`	a modhash

`upload_type`	one of (`img`, `header`, `icon`, `banner`)

GET /api/v1/*subreddit*/post_requirements submit

Fetch moderator-designated requirements to post to the subreddit.

Moderators may enable certain restrictions, such as minimum title length, when making a submission to their subreddit.

Clients may use the values returned by this endpoint to pre-validate fields before making a request to POST /api/submit. This may allow the client to provide a better user experience to the user, for example by creating a text field in their app that does not allow the user to enter more characters than the max title length.

A non-exhaustive list of possible requirements a moderator may enable:

- `body_blacklisted_strings` : List of strings. Users may not submit posts that contain these words.
- `body_restriction_policy` : String. One of "required", "notAllowed", or "none", meaning that a self-post body is required, not allowed, or optional, respectively.
- `domain_blacklist` : List of strings. Users may not submit links to these domains
- `domain_whitelist` : List of strings. Users submissions MUST be from one of these domains
- `is_flair_required` : Boolean. If True, flair must be set at submission time.
- `title_blacklisted_strings` : List of strings. Submission titles may NOT contain any of the listed strings.
- `title_required_strings` : List of strings. Submission title MUST contain at least ONE of the listed strings.
- `title_text_max_length` : Integer. Maximum length of the title field.
- `title_text_min_length` : Integer. Minimum length of the title field.

GET /r/*subreddit*/about read

Return information about the subreddit.

Data includes the subscriber count, description, and header image.

GET /r/*subreddit*/about/edit modconfig

Get the current settings of a subreddit.

In the API, this returns the current settings of the subreddit as used by /api/site_admin. On the HTML site, it will display a form for editing the subreddit.

`created`	one of (`true`, `false`)
`location`	

GET /r/*subreddit*/about/rules read

Get the rules for the current subreddit

GET /r/*subreddit*/about/traffic modconfig

GET [/r/*subreddit*]/sidebar read

Get the sidebar for the current subreddit

GET [/r/*subreddit*]/sticky read

Redirect to one of the posts stickied in the current subreddit

The "num" argument can be used to select a specific sticky, and will default to 1 (the top sticky) if not specified. Will 404 if there is not currently a sticky post in this subreddit.

num	an integer between 1 and 2 (default: 1)

GET **/subreddits/mine/*where*** mysubreddits rss support

→ **/subreddits/mine/subscriber**
→ **/subreddits/mine/contributor**
→ **/subreddits/mine/moderator**
→ **/subreddits/mine/streams**

Get subreddits the user has a relationship with.

The `where` parameter chooses which subreddits are returned as follows:

- `subscriber` - subreddits the user is subscribed to
- `contributor` - subreddits the user is an approved user in
- `moderator` - subreddits the user is a moderator of
- `streams` - subscribed to subreddits that contain hosted video links

See also: /api/subscribe, /api/friend, and /api/accept_moderator_invite.

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
show	(optional) the string `all`
sr_detail	(optional) expand subreddits

GET **/subreddits/search** read rss support

Search subreddits by title and description.

This endpoint is a listing.

after	fullname of a thing
before	fullname of a thing
count	a positive integer (default: 0)
limit	the maximum number of items desired (default: 25, maximum: 100)
q	a search query
search_query_id	a uuid
show	(optional) the string `all`
show_users	boolean value
sort	one of (`relevance`, `activity`)
sr_detail	(optional) expand subreddits
typeahead_active	boolean value or None

GET **/subreddits/*where*** read rss support

→ **/subreddits/popular**
→ **/subreddits/new**
→ **/subreddits/gold**
→ **/subreddits/default**

Get all subreddits.

The `where` parameter chooses the order in which the subreddits are displayed. `popular` sorts on the activity of the subreddit and the position of the subreddits can shift around. `new` sorts the subreddits based on their creation date, newest first.

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

GET **/users/search** read rss support

Search user profiles by title and description.

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`q`	a search query
`search_query_id`	a uuid
`show`	(optional) the string `all`
`sort`	one of (`relevance`, `activity`)
`sr_detail`	(optional) expand subreddits
`typeahead_active`	boolean value or None

GET **/users/*where*** read rss support

→ **/users/popular**
→ **/users/new**

Get all user subreddits.

The `where` parameter chooses the order in which the subreddits are displayed. `popular` sorts on the activity of the subreddit and the position of the subreddits can shift around. `new` sorts the user subreddits based on their creation date, newest first.

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

users

POST **/api/block_user** account

For blocking a user. Only accessible to approved OAuth applications

`account_id`	fullname of a account
`api_type`	the string `json`
`name`	A valid, existing reddit username
`uh / X-Modhash header`	a modhash

POST **[/r/*subreddit*]/api/friend** any

Create a relationship between a user and another user or subreddit

OAuth2 use requires appropriate scope based on the 'type' of the relationship:

- moderator: Use "moderator_invite"
- moderator_invite: `modothers`
- contributor: `modcontributors`
- banned: `modcontributors`
- muted: `modcontributors`
- wikibanned: `modcontributors` and `modwiki`
- wikicontributor: `modcontributors` and `modwiki`
- friend: Use /api/v1/me/friends/{username}
- enemy: Use /api/block

Complement to POST_unfriend

`api_type`	the string `json`
`ban_context`	fullname of a thing
`ban_message`	raw markdown text
`ban_reason`	a string no longer than 100 characters
`container`	
`duration`	an integer between 1 and 999
`name`	the name of an existing user
`note`	a string no longer than 300 characters
`permissions`	
`type`	one of (`friend`, `moderator`, `moderator_invite`, `contributor`, `banned`, `muted`, `wikibanned`, `wikicontributor`)
`uh / X-Modhash header`	a modhash

POST **/api/report_user** report

Report a user. Reporting a user brings it to the attention of a Reddit admin.

`details`	JSON data
`reason`	a string no longer than 100 characters
`('user',)`	A valid, existing reddit username

POST **[/r/*subreddit*]/api/setpermissions** modothers

api_type	the string json
name	the name of an existing user
permissions	
type	
uh / X-Modhash header	a modhash

POST [/r/*subreddit*]/api/unfriend any

Remove a relationship between a user and another user or subreddit

The user can either be passed in by name (nuser) or by fullname (iuser). If type is friend or enemy, 'container' MUST be the current user's fullname; for other types, the subreddit must be set via URL (e.g., /r/funny/api/unfriend)

OAuth2 use requires appropriate scope based on the 'type' of the relationship:

- moderator: `modothers`
- moderator_invite: `modothers`
- contributor: `modcontributors`
- banned: `modcontributors`
- muted: `modcontributors`
- wikibanned: `modcontributors` and `modwiki`
- wikicontributor: `modcontributors` and `modwiki`
- friend: Use /api/v1/me/friends/{username}
- enemy: `privatemessages`

Complement to POST_friend

api_type	the string json
container	
id	fullname of a thing
name	the name of an existing user
type	one of (friend, enemy, moderator, moderator_invite, contributor, banned, muted, wikibanned, wikicontributor)
uh / X-Modhash header	a modhash

GET /api/user_data_by_account_ids privatemessages

ids	A comma-separated list of account fullnames

GET /api/username_available any

Check whether a username is available for registration.

user	a valid, unused, username

DELETE /api/v1/me/friends/*username* subscribe

Stop being friends with a user.

id	A valid, existing reddit username

GET /api/v1/me/friends/*username* mysubreddits

Get information about a specific 'friend', such as notes.

`id`	A valid, existing reddit username

PUT **/api/v1/me/friends/***username* subscribe

Create or update a "friend" relationship.

This operation is idempotent. It can be used to add a new friend, or update an existing friend (e.g., add/change the note on that friend)

`This endpoint expects JSON data of this format`	`{ "name": A valid, existing reddit username, "note": a string no longer than 300 characters, }`

GET **/api/v1/user/***username***/trophies** read

Return a list of trophies for the a given user.

`id`	A valid, existing reddit username

GET **/user/***username***/about** read

Return information about the user, including karma and gold status.

`username`	the name of an existing user

GET **/user/***username***/***where* history rss support

→ **/user/***username***/overview**
→ **/user/***username***/submitted**
→ **/user/***username***/comments**
→ **/user/***username***/upvoted**
→ **/user/***username***/downvoted**
→ **/user/***username***/hidden**
→ **/user/***username***/saved**
→ **/user/***username***/gilded**

This endpoint is a listing.

`context`	an integer between 2 and 10
`show`	one of (`given`)
`sort`	one of (`hot`, `new`, `top`, `controversial`)
`t`	one of (`hour`, `day`, `week`, `month`, `year`, `all`)
`type`	one of (`links`, `comments`)
`username`	the name of an existing user
`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`sr_detail`	(optional) expand subreddits

widgets

POST **[/r/***subreddit***]/api/widget** structuredstyles

Add and return a widget to the specified subreddit

Accepts a JSON payload representing the widget data to be saved. Valid payloads differ in shape based on the "kind" attribute passed on the root object, which must be a valid widget kind.

`json`

json data:

```
{
  "data": [
    {
      "height": an integer,
      "linkUrl": A valid URL (optional),
      "url": a valid URL of a reddit-
hosted image,
      "width": an integer,
    },
    ...
  ],
  "kind": one of (`image`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "configuration": {
    "numEvents": an integer between 1 and
50 (default: 10),
    "showDate": boolean value,
    "showDescription": boolean value,
    "showLocation": boolean value,
    "showTime": boolean value,
    "showTitle": boolean value,
  },
  "googleCalendarId": a valid email
address,
  "kind": one of (`calendar`),
  "requiresSync": boolean value,
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "kind": one of (`textarea`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
  "text": raw markdown text,
}

OR

{
  "data": [
    {
      "text": a string no longer than 20
characters,
      "url": a valid URL,
    }

    OR

    {
      "children": [
        {
          "text": a string no longer than
20 characters,
          "url": a valid URL,
        },
        ...
      ],
      "text": a string no longer than 20
characters,
    },
    ...
  ],
  "kind": one of (`menu`),
  "showWiki": boolean value,
}

OR

{
  "buttons": [
    {
      "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
      "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      "hoverState": {
        "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
        "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
        "kind": one of (`text`),
        "text": a string no longer than
30 characters,
        "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      }

      OR

      {
        "height": an integer,
        "imageUrl": a valid URL of a
reddit-hosted image,
        "kind": one of (`image`),
        "width": an integer,
      },
      "kind": one of (`text`),
      "text": a string no longer than 30
characters,
      "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      "url": a valid URL,
    }

    OR

    {
      "height": an integer,
      "hoverState": {
        "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
        "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
        "kind": one of (`text`),
        "text": a string no longer than
30 characters,
        "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      }

      OR

      {
        "height": an integer,
        "imageUrl": a valid URL of a
reddit-hosted image,
        "kind": one of (`image`),
        "width": an integer,
      },
      "imageUrl": a valid URL of a
reddit-hosted image,
      "kind": one of (`image`),
      "linkUrl": a valid URL,
      "text": a string no longer than 30
characters,
      "width": an integer,
    },
    ...
  ],
  "description": raw markdown text,
  "kind": one of (`button`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "data": [
    subreddit name,
    ...
  ],
  "kind": one of (`community-list`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "css": a string no longer than 100000
characters,
  "height": an integer between 50 and
500,
  "imageData": [
    {
      "height": an integer,
      "name": a string no longer than 20
characters,
      "url": a valid URL of a reddit-
hosted image,
      "width": an integer,
    },
    ...
  ],
  "kind": one of (`custom`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
  "text": raw markdown text,
}

OR

{
  "display": one of (`cloud`, `list`),
  "kind": one of (`post-flair`),
  "order": [
    a flair template ID,
    ...
  ],
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}
```

DELETE **[/r/*subreddit*]/api/widget/*widget_id*** structuredstyles

Delete a widget from the specified subreddit (if it exists)

`widget_id`	id of an existing widget

PUT **[/r/*subreddit*]/api/widget/*widget_id*** structuredstyles

Update and return the data of a widget.

Accepts a JSON payload representing the widget data to be saved. Valid payloads differ in shape based on the "kind" attribute passed on the root object, which must be a valid widget kind.

`json` json data:

```
{
  "data": [
    {
      "height": an integer,
      "linkUrl": A valid URL (optional),
      "url": a valid URL of a reddit-
hosted image,
      "width": an integer,
    },
    ...
  ],
  "kind": one of (`image`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "configuration": {
    "numEvents": an integer between 1 and
50 (default: 10),
    "showDate": boolean value,
    "showDescription": boolean value,
    "showLocation": boolean value,
    "showTime": boolean value,
    "showTitle": boolean value,
  },
  "googleCalendarId": a valid email
address,
  "kind": one of (`calendar`),
  "requiresSync": boolean value,
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "kind": one of (`textarea`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
  "text": raw markdown text,
}

OR

{
  "display": one of (`full`, `compact`),
  "kind": one of (`subreddit-rules`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "data": [
    {
      "text": a string no longer than 20
characters,
      "url": a valid URL,
    }

    OR

    {
      "children": [
        {
          "text": a string no longer than
20 characters,
          "url": a valid URL,
        },
        ...
      ],
      "text": a string no longer than 20
characters,
    },
    ...
  ],
  "kind": one of (`menu`),
  "showWiki": boolean value,
}

OR

{
  "buttons": [
    {
      "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
      "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      "hoverState": {
        "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
        "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
        "kind": one of (`text`),
        "text": a string no longer than
30 characters,
        "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
      }

      OR
```

```
   {
    "height": an integer,
    "imageUrl": a valid URL of a
reddit-hosted image,
    "kind": one of (`image`),
    "width": an integer,
   },
   "kind": one of (`text`),
   "text": a string no longer than 30
characters,
   "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
   "url": a valid URL,
  }

  OR

  {
   "height": an integer,
   "hoverState": {
    "color": a 6-digit rgb hex color,
e.g. `#AABBCC`,
    "fillColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "kind": one of (`text`),
    "text": a string no longer than
30 characters,
    "textColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
   }

   OR

   {
    "height": an integer,
    "imageUrl": a valid URL of a
reddit-hosted image,
    "kind": one of (`image`),
    "width": an integer,
   },
   "imageUrl": a valid URL of a
reddit-hosted image,
   "kind": one of (`image`),
   "linkUrl": a valid URL,
   "text": a string no longer than 30
characters,
   "width": an integer,
  },
  ...
 ],
 "description": raw markdown text,
 "kind": one of (`button`),
 "shortName": a string no longer than 30
characters,
 "styles": {
  "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
 },
}

OR

{
 "currentlyViewingText": a string no
longer than 30 characters,
 "kind": one of (`id-card`),
 "shortName": a string no longer than 30
characters,
 "styles": {
  "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
 },
 "subscribersText": a string no longer
than 30 characters,
}
```

```
OR

{
  "data": [
    subreddit name,
    ...
  ],
  "kind": one of (`community-list`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "css": a string no longer than 100000
characters,
  "height": an integer between 50 and
500,
  "imageData": [
    {
      "height": an integer,
      "name": a string no longer than 20
characters,
      "url": a valid URL of a reddit-
hosted image,
      "width": an integer,
    },
    ...
  ],
  "kind": one of (`custom`),
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
  "text": raw markdown text,
}

OR

{
  "display": one of (`cloud`, `list`),
  "kind": one of (`post-flair`),
  "order": [
    a flair template ID,
    ...
  ],
  "shortName": a string no longer than 30
characters,
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}

OR

{
  "kind": one of (`moderators`),
  "styles": {
    "backgroundColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
    "headerColor": a 6-digit rgb hex
color, e.g. `#AABBCC`,
  },
}
```

`widget_id`	a valid widget id

POST **[/r/*subreddit*]/api/widget_image_upload_s3**

structuredstyles

Acquire and return an upload lease to s3 temp bucket.

The return value of this function is a json object containing credentials for uploading assets to S3 bucket, S3 url for upload request and the key to use for uploading. Using this lease the client will upload the emoji image to S3 temp bucket (included as part of the S3 URL).

This lease is used by S3 to verify that the upload is authorized.

`filepath`	name and extension of the image file e.g. image1.png
`mimetype`	mime type of the image e.g. image/png

PATCH **[/r/*subreddit*]/api/widget_order/*section***

structuredstyles

Update the order of widget_ids in the specified subreddit

`json`	json data:
	`[`
	`  a string,`
	`  ...`
	`]`

`section`	one of (`sidebar`)

GET **[/r/*subreddit*]/api/widgets** structuredstyles

Return all widgets for the given subreddit

`progressive_images`	boolean value

wiki

POST **[/r/*subreddit*]/api/wiki/alloweditor/*act*** modwiki

→ **[/r/*subreddit*]/api/wiki/alloweditor/del**
→ **[/r/*subreddit*]/api/wiki/alloweditor/add**

Allow/deny `username` to edit this wiki `page`

`act`	one of (`del`, `add`)
`page`	the name of an existing wiki page
`uh / X-Modhash header`	a modhash
`username`	the name of an existing user

POST **[/r/*subreddit*]/api/wiki/edit** wikiedit

Edit a wiki `page`

`content`	
`page`	the name of an existing page or a new page to create
`previous`	the starting point revision for this edit
`reason`	a string up to 256 characters long, consisting of printable characters.
`uh / X-Modhash`	a modhash

header

POST [/r/*subreddit*]/api/wiki/hide modwiki

Toggle the public visibility of a wiki page revision

`page`	the name of an existing wiki page
`revision`	a wiki revision ID
`uh / X-Modhash header`	a modhash

POST [/r/*subreddit*]/api/wiki/revert modwiki

Revert a wiki `page` to `revision`

`page`	the name of an existing wiki page
`revision`	a wiki revision ID
`uh / X-Modhash header`	a modhash

GET [/r/*subreddit*]/wiki/discussions/*page* wikiread

Retrieve a list of discussions about this wiki `page`

This endpoint is a listing.

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`page`	the name of an existing wiki page
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

GET [/r/*subreddit*]/wiki/pages wikiread

Retrieve a list of wiki pages in this subreddit

GET [/r/*subreddit*]/wiki/revisions wikiread

Retrieve a list of recently changed wiki pages in this subreddit

`after`	fullname of a thing
`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

GET [/r/*subreddit*]/wiki/revisions/*page* wikiread

Retrieve a list of revisions of this wiki `page`

This endpoint is a listing.

`after`	fullname of a thing

`before`	fullname of a thing
`count`	a positive integer (default: 0)
`limit`	the maximum number of items desired (default: 25, maximum: 100)
`page`	the name of an existing wiki page
`show`	(optional) the string `all`
`sr_detail`	(optional) expand subreddits

GET **[/r/*subreddit*]/wiki/settings/*page*** modwiki

Retrieve the current permission settings for `page`

`page`	the name of an existing wiki page

POST **[/r/*subreddit*]/wiki/settings/*page*** modwiki

Update the permissions and visibility of wiki `page`

`listed`	boolean value
`page`	the name of an existing wiki page
`permlevel`	an integer
`uh / X-Modhash header`	a modhash

GET **[/r/*subreddit*]/wiki/*page*** wikiread

Return the content of a wiki page

If `v` is given, show the wiki page as it was at that version If both `v` and `v2` are given, show a diff of the two

`page`	the name of an existing wiki page
`v`	a wiki revision ID
`v2`	a wiki revision ID

about

blog
about
advertising
careers

help

site rules
Reddit help center
reddiquette
mod guidelines
contact us

apps & tools

Reddit for iPhone
Reddit for Android
mobile website

<3

reddit premium

π