This repository has been archived by the owner before Nov 9, 2022. It is now read-only.

reddit-archive / **reddit** Public archive

Code Pull requests Actions Projects **Wiki** Security Insights

OAuth2

Jump to bottom

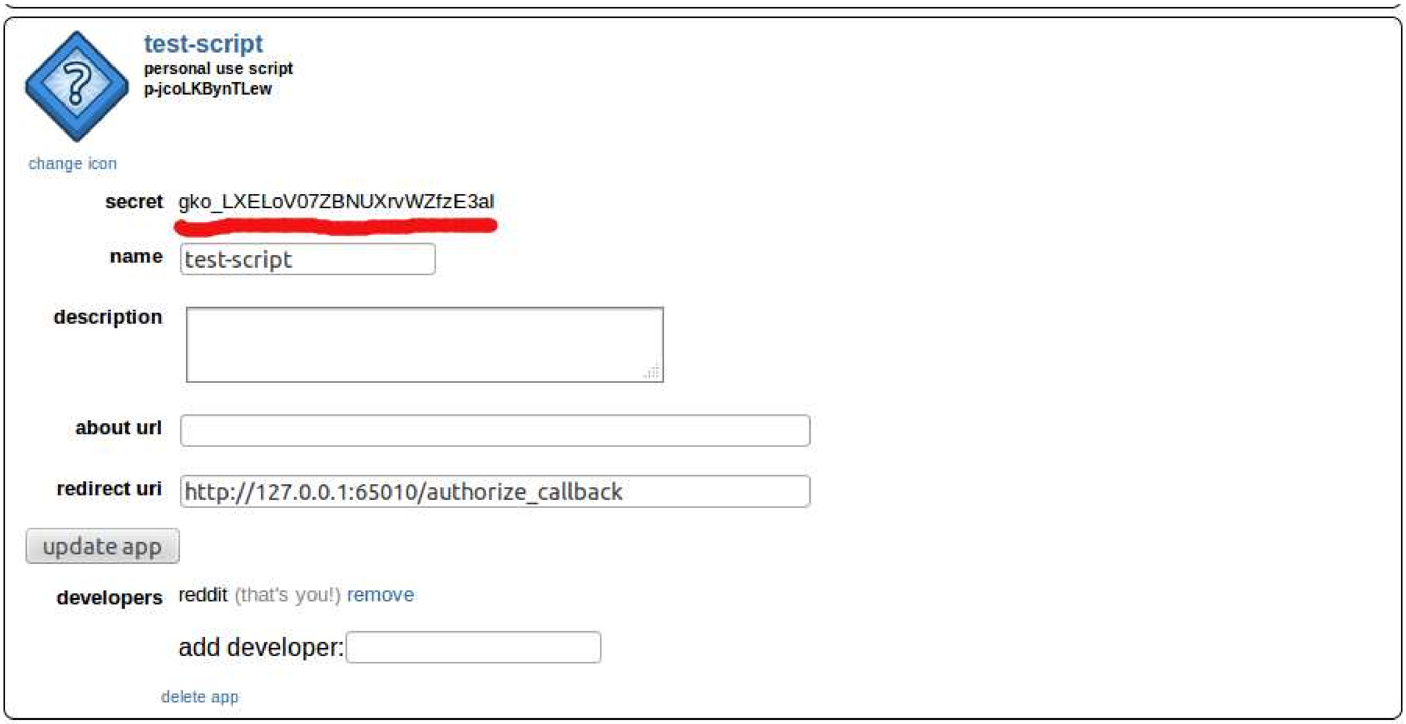
Kevin M Granger edited this page on Jan 8, 2016 · 49 revisions

OAuth2 support allows you to use reddit to authenticate on non-reddit websites and applications.

Looking to jump right in? See the Quick Start Example, for bots and personal scripts only

Getting Started

First you need an application id and secret so reddit knows your application. You get this information by going to https://www.reddit.com/prefs/apps and clicking "are you a developer? create an app..."



test-script
personal use script
p-jcoLKBynTLew

secret gko_LXELoV07ZBNUXrvWZfzE3al

name test-script

description

about url

redirect uri http://127.0.0.1:65010/authorize_callback

developers reddit (that's you!)

add developer:

When registering your app, it's important to choose the correct and relevant app "type," as the type determines what authentication paths your app may take. Read more on app types.

- Web app: Runs as part of a web service on a server you control. Can keep a secret.
- Installed app: Runs on devices you don't control, such as the user's mobile phone. *Cannot* keep a secret, and therefore, does not receive one.
- Script app: Runs on hardware you control, such as your own laptop or server. Can keep a secret. Only has access to your account.

Be sure to give the app a reasonable name and description. The redirect uri is important - for web apps, it points to a URL on a webserver that you control.

The part underlined in red is your client secret. *You should never share this.* Non-confidential clients (installed apps) *do not* have a secret.

Authorization

In order to make requests to reddit's API via OAuth, you must acquire an Authorization token, either on behalf of a user or for your client (see Application Only OAuth, below). To act on behalf of a user, the user has to let reddit.com know that they're ok with your app performing certain actions for them, such as reading their subreddit subscriptions or sending a private message. In order to do so, your website or app should send the user to the authorization URL:

```
https://www.reddit.com/api/v1/authorize?client_id=CLIENT_ID&response_type=TYPE&
    state=RANDOM_STRING&redirect_uri=URI&duration=DURATION&scope=SCOPE_STRING
```

Note: Use `/api/v1/authorize.compact?` for a page that's friendlier to small screens.

Parameter	Values	Description
`client_id`	The Client ID generated during app registration	Tells reddit.com which app is making the request
`response_type`	`code`	Must be the string "code". For implicit grants, see below.
`state`	A string of your choosing	You should generate a unique, possibly random, string for each authorization request. This value will be returned to you when the user visits your REDIRECT_URI after allowing your app access - you should verify that it matches the one you sent. This ensures that only authorization requests you've started are ones you finish. (You may also use this

Parameter	Values	Description
		value to, for example, tell your webserver what action to take after receiving the OAuth2 bearer token)
`redirect_uri`	The redirect_uri you have specified during registration	If this does not match the registered redirect_uri, the authorization request will fail. If authorization succeeds, the user's browser will be instructed to redirect to this location.
`duration`	Either `temporary` or `permanent`	Indicates whether or not your app needs a permanent token. All bearer tokens expire after 1 hour. If you indicate you need `permanent` access to a user's account, you will additionally receive a `refresh_token` when acquiring the bearer token. You may use the `refresh_token` to acquire a new bearer token after your current token expires. Choose `temporary` if you're completing a one-time request for the user (such as analyzing their recent comments); choose `permanent` if you will be performing ongoing tasks for the user, such as notifying them whenever they receive a private message. **The implicit grant flow does not allow permanent tokens.**
`scope`	A space-separated* list of scope strings	All bearer tokens are limited in what functions they may perform. You must explicitly request access to areas of the api, such as private messaging or moderator actions. See our automatically generated API docs. Scope Values: `identity`, `edit`, `flair`, `history`, `modconfig`, `modflair`, `modlog`, `modposts`, `modwiki`, `mysubreddits`, `privatemessages`, `read`, `report`, `save`, `submit`, `subscribe`, `vote`, `wikiedit`, `wikiread`.

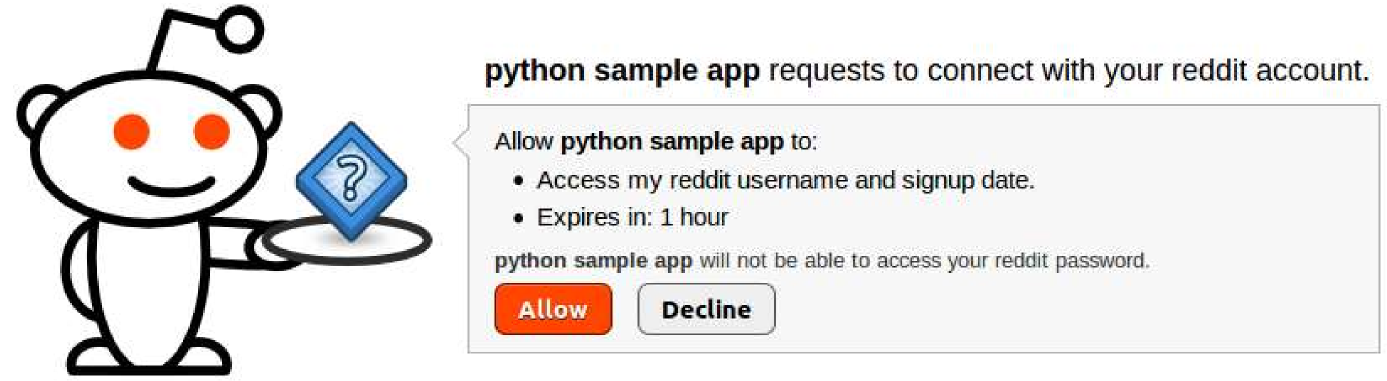
See http://www.reddit.com/dev/api/oauth for a breakdown of which API endpoints require which scopes. Also see https://www.reddit.com/api/v1/scopes for a list of all available scopes.

Error	Cause	Resolution
User sees 403 error in browser	client_id is missing or invalid	Verify that client_id is set and correct for your app

Error	Cause	Resolution
User sees 403 error in browser	redirect_uri is invalid	Verify that redirect_uri is set and matches what is set for your app

* *Note*: Commas are supported too.

When you send the user to the authorization URL, they will be shown what parts of their account you want access to based on the requested scopes:



python sample app requests to connect with your reddit account.

Allow **python sample app** to:

- Access my reddit username and signup date.
- Expires in: 1 hour

python sample app will not be able to access your reddit password.

You will only be able to acquire a bearer token if the user decides they trust your app with the permissions (scopes) you've requested, so be sure to limit your permission request to only those that encompass the API endpoints you required.

Token Retrieval (`code` flow)

Allowing the user to authorize your application

If the user chooses to allow your application, their browser will be instructed to redirect to your app's registered `redirect_uri` . The redirect URI will have the information below attached as query parameters. You should parse the query parameters for the URI for use in the next step.

Parameter	Values	Description
`error`	`access_denied` , others	See error table below for list of causes.
`code`	A string	A one-time use code that may be exchanged for a bearer token. See the next step

Parameter	Values	Description
`state`	A string	This value should be the same as the one sent in the initial authorization request, and **your app should verify that it is, in fact, the same**. Your app may also do anything else it wishes with the state info, such as parse a portion of it to determine what action to perform on behalf of the user.

Error	Cause	Resolution
`access_denied`	User chose not to grant your app permissions	Fail gracefully - let the user know you cannot continue, and be respectful of their choice to decline to use your app
`unsupported_response_type`	Invalid `response_type` parameter in initial Authorization	Ensure that the `response_type` parameter is one of the allowed values
`invalid_scope`	Invalid `scope` parameter in initial Authorization	Ensure that the `scope` parameter is a space-separated list of valid scopes
`invalid_request`	There was an issue with the request sent to `/api/v1/authorize`	Double check the parameters being sent during the request to `/api/v1/authorize` above.

Retrieving the access token

If you didn't get an `error` and the `state` value checks out, you may then make a POST request with `code` to the following URL to retrieve your access token:

```
https://www.reddit.com/api/v1/access_token
```

Include the following information in your POST data (NOT as part of the URL)

```
grant_type=authorization_code&code=CODE&redirect_uri=URI
```

Header	Values	Description
Authorization	HTTP Basic Auth	The "user" is the `client_id`. The "password" for confidential clients is the `client_secret`. The "password" for non-confidential clients (installed apps) is an empty string.

Parameter	Values	Description
`grant_type`	`authorization_code`	Indicates that you're using the "standard" code based flow. Other values not relevant to this flow are `refresh_token` (for renewing an access token) and `password` (for script apps only)
`code`	A string	The `code` your app retrieved above
`redirect_uri`	The redirect_uri registered to your app	Yes, you need it here again, and yes, it must match exactly.

Error	Cause	Resolution
401 response	Client credentials sent as HTTP Basic Authorization were invalid	Verify that you are properly sending HTTP Basic Authorization headers and that your credentials are correct
`unsupported_grant_type`	`grant_type` parameter was invalid or Http Content type was not set correctly	Verify that the `grant_type` sent is supported and make sure the content type of the http message is set to `application/x-www-form-urlencoded`
`NO_TEXT` for field `code`	You didn't include the `code` parameter	Include the `code` parameter in the POST data
`invalid_grant`	The `code` has expired or already been used	Ensure that you are not attempting to re-use old `code` s - they are one time use.

The response from this request, if successful, will be JSON of the following format:

```
{
    "access_token": Your access token,
    "token_type": "bearer",
    "expires_in": Unix Epoch Seconds,
    "scope": A scope string,
    "refresh_token": Your refresh token
}
```

"refresh_token" will only be present if one was requested. You may now make API requests to reddit's servers on behalf of that user, by including the following header in your HTTP requests:

```
Authorization: bearer TOKEN
```

API requests with a bearer token should be made to `https://oauth.reddit.com`, NOT www.reddit.com.

Refreshing the token

Access tokens expire after one hour. If your app requires access after that time, it must request a refresh token by including `duration=permanent` with the authorization request (see above). When the current access token expires, your app should send another POST request to the access token URL:

```
https://www.reddit.com/api/v1/access_token
```

Include the following information in your POST data (NOT as part of the URL)

```
grant_type=refresh_token&refresh_token=TOKEN
```

Header	Values	Description
Authorization	HTTP Basic Auth	Must be the same HTTP Basic Authentication as when requesting the original refresh token.

Parameter	Values	Description
`grant_type`	`refresh_token`	Indicates that you're requesting a new access token using a refresh token
`refresh_token`	A string	The refresh token retrieved during the initial request for an access token

Error	Cause	Resolution
401 response	Client credentials sent as HTTP Basic Authorization were invalid	Verify that you are properly sending HTTP Basic Authorization headers and that your credentials are correct
`unsupported_grant_type`	`grant_type` parameter was invalid	Verify that the `grant_type` sent is supported
`NO_TEXT` for field `refresh_token`	`refresh_token` parameter was missing	Include a valid `refresh_token` in the POST parameters

The response will look the same as the initial access token request.

```
{
    "access_token": Your access token,
    "token_type": "bearer",
    "expires_in": Unix Epoch Seconds,
    "scope": A scope string,
}
```

Authorization (Implicit grant flow)

Note: *Only apps created as "installed" type apps may use the implicit flow. "web" and "script" type apps are considered "confidential" (i.e., they have secrets). Since you cannot safely send a secret via the implicit flow, we have elected to disallow implicit access to apps with secrets.*

In order to make requests to reddit's API via OAuth, you must acquire an Authorization token, either on behalf of a user or for your client (see Application Only OAuth, below). To act on behalf of a user, the user has to let reddit.com know that they're ok with your app performing certain actions for them, such as reading their subreddit subscriptions or sending a private message. In order to do so, your website or app should send the user to the authorization URL:

```
https://www.reddit.com/api/v1/authorize?client_id=CLIENT_ID&response_type=TYPE&
    state=RANDOM_STRING&redirect_uri=URI&scope=SCOPE_STRING
```

Parameter	Values	Description
`client_id`	The Client ID generated during app registration	Tells reddit.com which app is making the request
`response_type`	`token`	Must be the string "token".
`state`	A string of your choosing	You should generate a unique, possibly random, string for each authorization request. This value will be returned to you when the user visits your REDIRECT_URI after allowing your app access - you should verify that it matches the one you sent. This ensures that only authorization requests you've started are ones you finish. (You may also use this value to, for example, tell your webserver what action to take after receiving the OAuth2 bearer token)
`redirect_uri`	The redirect_uri you have specified during registration	If this does not match the registered redirect_uri, the authorization request will fail. If authorization succeeds, the user's browser will be instructed to redirect to this location.
`scope`	A space-separated* list of scope strings	All bearer tokens are limited in what functions they may perform. You must explicitly request access to areas of the api, such as private messaging or moderator actions. See our automatically generated API docs. Scope Values: `identity` , `edit` , `flair` , `history` , `modconfig` , `modflair` , `modlog` , `modposts` , `modwiki` , `mysubreddits` , `privatemessages` , `read` , `report` , `save` , `submit` , `subscribe` , `vote` , `wikiedit` , `wikiread` .

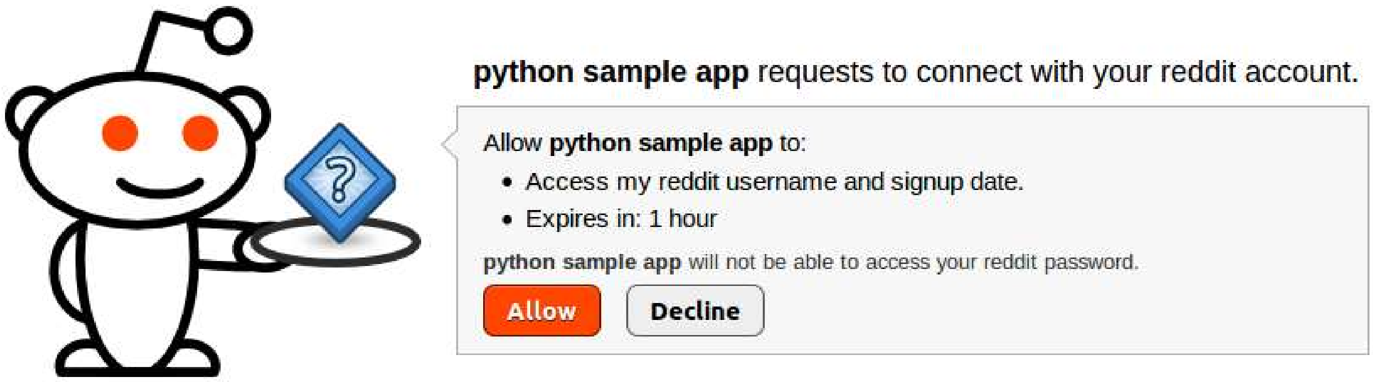
See http://www.reddit.com/dev/api/oauth for a breakdown of which API endpoints require which scopes. Also see https://www.reddit.com/api/v1/scopes for a list of all available scopes.

Error	Cause	Resolution
User sees 403 error in browser	client_id is missing or invalid	Verify that client_id is set and correct for your app

Error	Cause	Resolution
User sees 403 error in browser	redirect_uri is invalid	Verify that redirect_uri is set and matches what is set for your app

* *Note*: This is a slight deviation from the OAuth 2.0 specification, which states scopes should normally be space-separated.

When you send the user to the authorization URL, they will be shown what parts of their account you want access to based on the requested scopes:



python sample app requests to connect with your reddit account.

Allow **python sample app** to:

- Access my reddit username and signup date.
- Expires in: 1 hour

python sample app will not be able to access your reddit password.

You will only be able to acquire a bearer token if the user decides they trust your app with the permissions (scopes) you've requested, so be sure to limit your permission request to only those that encompass the API endpoints you required.

Token Retrieval (Implicit grant flow)

If the user chooses to allow your application, their browser will be instructed to redirect to your app's registered `redirect_uri` . The redirect URI will have the information below encoded into the fragment portion of the URL.

Error	Cause	Resolution
`access_denied`	User chose not to grant your app permissions	Fail gracefully - let the user know you cannot continue, and be respectful of their choice to decline to use your app
`unsupported_response_type`	Invalid `response_type` parameter in initial	Ensure that the `response_type` parameter is one of the allowed

Error	Cause	Resolution
	Authorization	values
`invalid_scope`	Invalid `scope` parameter in initial Authorization	Ensure that the `scope` parameter is a space-separated list of valid scopes
`invalid_request`	There was an issue with the request sent to `/api/v1/authorize`	Double check the parameters being sent during the request to `/api/v1/authorize` above.

The response from this request, if successful, will be form encoded into the fragment with the following values:

Parameter	Explanation
access_token	Your access token,
token_type	The string "bearer"
expires_in	Seconds until the token expires
scope	The scope of the token
state	This value should be the same as the one sent in the initial authorization request, and **your app should verify that it is, in fact, the same**. Your app may also do anything else it wishes with the state info, such as parse a portion of it to determine what action to perform on behalf of the user.

You may now make API requests to reddit's servers on behalf of that user, by including the following header in your HTTP requests:

```
Authorization: bearer TOKEN
```

API requests with a bearer token should be made to `https://oauth.reddit.com` , NOT www.reddit.com.

Manually Revoking a Token

While access tokens expire after 1 hour, and the end user can always revoke a client's tokens, good clients still clean up after themselves. OAuth2 clients can manually revoke tokens they are finished with - useful for ensuring that tokens, if stolen, aren't usable, and just for acting as a good citizen when the user "logs out" of your website (as an example).

When a client is completely done with a token, it can POST to the following URL to permanently revoke the token:

```
https://www.reddit.com/api/v1/revoke_token
```

Include the following information in your POST data (not in the URL):

```
token=TOKEN&token_type_hint=TOKEN_TYPE
```

Your client must be authenticated using HTTP Basic Authentication in the same manner as when requesting the original token.

Revoking a refresh token will also revoke any related access tokens!

Parameter	Values	Description
`token`	A refresh or access token	The access token or refresh token that the client wishes to revoke
`token_type_hint`	`refresh_token` or `access_token`	(optional) The type of token being revoked. If not included, the request will still succeed per normal, though may be slower.

Error	Cause	Resolution
401 response	Client credentials sent as HTTP Basic Authorization were invalid	Verify that you are properly sending HTTP Basic Authorization headers and that your credentials are correct

Note: Per RFC 7009, this request will return a success (204) response even if the passed in `token` was never valid.

Application Only OAuth

In some cases, 3rd party app clients may wish to make API requests without a user context. App clients can request a "user-less" Authorization token via either the standard `client_credentials` grant, or the reddit specific extension to this grant, `https://oauth.reddit.com/grants/installed_client` . Which grant type an app uses depends on the app-type and its use case:

- `https://oauth.reddit.com/grants/installed_client` :
  - Installed app types (as these apps are considered "non-confidential", have no secret, and thus, are ineligible for `client_credentials` grant.
  - Other apps acting on behalf of one or more "logged out" users.
- `client_credentials` :
  - Confidential clients (web apps / scripts) not acting on behalf of one or more logged out users.

For both grant types, the app should make a request to the following endpoint to retrieve your access token:

```
https://www.reddit.com/api/v1/access_token
```

For `client_credentials` grants include the following information in your POST data (NOT as part of the URL)

```
grant_type=client_credentials
```

When using the `https://oauth.reddit.com/grants/installed_client` grant, include the following information in your POST data:

```
grant_type=https://oauth.reddit.com/grants/installed_client&\
device_id=DEVICE_ID
```

You must supply your OAuth2 client's credentials via HTTP Basic Auth for this request. The "user" is the `client_id` , the "password" is the `client_secret` .

Parameter	Values	Descr
`grant_type`	`client_credentials` or `https://oauth.reddit.com/grants/installed_client`	The Application-only OAu requesting a token for
`device_id`	20-30 character ASCII string	A unique, per-device ID ge See note below. Only for

Parameter	Values	Descr
		`https://oauth.reddit.com` requests.

The response from this request, if successful, will be JSON of the following format:

```
{
    "access_token": Your access token,
    "token_type": "bearer",
    "expires_in": Unix Epoch Seconds,
    "scope": A scope string,
}
```

App-only OAuth token requests never receive a refresh_token.

What value should I use for `device_id` ?

You should generate and save unique ID on your client. The ID should be unique **per-device** or **per-user** of your app. A randomized or pseudo-randomized value is acceptable for generating the ID; however, you should retain and re-use the same device_id when renewing your access token. For example:

iOS:

```
NSString* uuid = [[NSUUID UUID] UUIDString];
```

Android:

```
import java.util.UUID;
String uuid = UUID.randomUUID().toString();
```

DO NOT use any personally identifiable information (including non-user-resettable information, such as Android's TelephonyManager.getDeviceId() or Apple's IDFA).

reddit *may* choose to use this ID to generate aggregate data about user counts. Clients that wish to remain anonymous should use the value `DO_NOT_TRACK_THIS_DEVICE` .

Why should I bother with all this, when I can just request API data for logged out users via https://www.reddit.com?

Great question! For one, using app-only OAuth may allow you to simplify your code - you can just always hit `https://oauth.reddit.com` with an `Authorization` header, rather than changing domains based on logged-out status. The other reason to use app-only OAuth is to access OAuth-only endpoints, such as /api/v1/user/{username}/trophies.

Examples

Here is example code on using OAuth in various languages:

- Simple Script Example
- Python Web Server Example
- Native iOS Example
- Python (with PRAW)
- PHP
- Java Server Example
- Android Example

What can I do once I have access?

You can look at the reddit API documentation page to see if a particular API has OAuth support and what scope it requires.

Other resources

Be sure to watch /r/redditdev for important OAuth 2 related announcements.

Pages 27

Find a page...

- Home
- API
- API Wrappers
- API: submit
- Architecture Overview

Show 12 more pages…

Clone this wiki locally

```
https://github.com/reddit-archive/reddit.wiki.git
```