

Forum:[Hack Forums](#)

Thread: [HF API Documentation](#)

Oh_Well - October 21, 2025, 2:51 pm

HF API(v2) Documentation

This is my 500th post and 30th thread for October.

The perfect timing to get my Pumpkin award while giving back to the community!

Last week, I started working on a project that relies on the Hack Forums API.

As a developer who's used to clean and structured documentation, I found it a bit challenging to navigate the threads discussing HF API v2.

So, I built a proper documentation page.

It covers everything needed to get started with developer apps, including the OAuth flow, permission scopes, and practical examples for both reading and writing data using JSON-based asks objects.

> apidocs.hackforums.net <

Skills.md

Spoiler ([Click to View](#))

Code HF API v2 Skills

=====

This file describes how an AI agent can understand and work with the **Hack Forums API v2** to help users design, debug, or reason about integrations. [hfapidocs](https://apidocs.hackforums.net/)

Core Concept

- HF API v2 is an OAuth-based HTTP API that exposes read and write operations for Hack Forums resources (users, posts, threads, bytes, contracts, etc.) via a unified `/read` and `/write` model plus helper endpoints. [hfapidocs](https://apidocs.hackforums.net/)

Base Details

- Base URL: `https://hackforums.net/api/v2`. [hfapidocs](https://apidocs.hackforums.net/)
- Main methods: `POST` with JSON bodies using an `asks` object to describe requested resources and fields. [hfapidocs](https://apidocs.hackforums.net/)
- Authentication: `Authorization: Bearer ACCESS\_TOKEN` header with OAuth 2.0 access tokens. [hfapidocs](https://apidocs.hackforums.net/)

OAuth & App Setup

The agent should know how to describe and reason about the OAuth flow, but not execute it itself.

- Developer apps are created in the HF Developer Portal (`usercp.php?action=apideveloper`) where you define name, description, redirect URI, details, and requested permissions (scopes).

[hfapidocs](https://apidocs.hackforums.net/)

- After approval, you receive a `client_id` and `client_secret` used in the OAuth authorization code flow. [hfapidocs](https://apidocs.hackforums.net/)

- OAuth steps: [hfapidocs](https://apidocs.hackforums.net/)

1. Redirect user to HF authorize page with `client_id`, `response_type=code`, optional `state`.
2. User grants scopes; HF redirects back to your redirect URI with `code` and `state` (if supplied).
3. Exchange `code` for an access token via `POST https://hackforums.net/api/v2/authorize` using form-encoded body (`grant_type=authorization_code`, `client_id`, `client_secret`, `code`).
4. Store the `access_token` securely and send it as a Bearer token on API calls.

Security constraints the agent should remind users of:

- Never put `client_secret` or access tokens in front-end or public code.

[hfapidocs](https://apidocs.hackforums.net/)

- Rotate tokens if compromised and handle revocation and 401 responses gracefully.

[hfapidocs](https://apidocs.hackforums.net/)

Scopes & Permissions

The agent should help users choose minimal scopes and understand what each scope unlocks.

[hfapidocs](https://apidocs.hackforums.net/)

Main scopes (during beta):

- Basic Info – public profile of the authorized user (e.g., `uid`, `username`, `usergroup`).

[hfapidocs](https://apidocs.hackforums.net/)

- Advanced Info – private profile details of the authorized user (e.g., `unreadpms`, `invisible`, `totalpms`). [hfapidocs](https://apidocs.hackforums.net/)

- Posts – access to forums, threads, posts; includes optional write permissions.

[hfapidocs](https://apidocs.hackforums.net/)

- Users – read public info of other members (`uid`, `username`, `avatar`, etc.).

[hfapidocs](https://apidocs.hackforums.net/)

- Bytes – read Byte logs plus optional write actions (transfers, deposits, withdrawals, bumps).

[hfapidocs](https://apidocs.hackforums.net/)

- Contracts – access contracts, disputes, and b-ratings. [hfapidocs](https://apidocs.hackforums.net/)

Scope changes:

- Increasing permissions requires users to re-authorize the application.

[hfapidocs](https://apidocs.hackforums.net/)

Reading Data (`/read`)

General pattern:

- Endpoint: `POST /read` or resource-specific helpers like `/read/posts`, `/read/me`, etc.

[hfapidocs](https://apidocs.hackforums.net/)

- Body structure uses an `asks` object; inputs to resources are prefixed with `_` (e.g., `_pid`, `_tid`).

[hfapidocs](https://apidocs.hackforums.net/)

- Fields set to `true` are requested back in the response.

[hfapidocs](https://apidocs.hackforums.net/)

Example patterns the agent should be able to describe or generate:

- Fetch current user data plus a specific thread: [hfapidocs](https://apidocs.hackforums.net/)
 - Use `/read` with `asks.me` for user fields (e.g., `uid`, `username`, `usergroup`) and `asks.threads` with `_tid` and nested fields like `firstpost.message` and author info.
- Fetch posts by `pid`: [hfapidocs](https://apidocs.hackforums.net/)
 - Use `/read/posts` with `asks.posts._pid` as a list and booleans for desired post fields (e.g., `pid`, `message`, `subject`, `dateline`, etc.).

Key read endpoints the agent should recognize: [hfapidocs](https://apidocs.hackforums.net/)

- `POST /read` – generic nested reads.
 - `POST /read/me` – current authorized user profile.
 - `POST /read/users` – public info of other users.
 - `POST /read/forums` – forum data.
 - `POST /read/threads` – threads by `tid`.
 - `POST /read/posts` – posts by `pid`.
 - `POST /read/bytes` – Byte logs.
 - `POST /read/contracts` – contracts data.
 - `POST /read/disputes` – disputes data.
 - `POST /read/bratings` – b-ratings.
 - `POST /read/sigmarket/market` and `POST /read/sigmarket/order` – signature market resources.
- [hfapidocs](https://apidocs.hackforums.net/)

Writing Data (`/write`)

General pattern:

- Endpoint: `POST /write` or resource-specific helpers such as `/write/posts`, `/write/bytes`, `/write/bytes/deposit`, `/write/bytes/withdraw`, `/write/bytes/bump`.

[hfapidocs](https://apidocs.hackforums.net/)

- Requires corresponding write scope for the resource (e.g., Posts, Bytes).

[hfapidocs](https://apidocs.hackforums.net/)

- Body uses an `asks` object with underscore-prefixed input fields (e.g., `_tid`, `_message`, `_amount`, `_to_uid`). [hfapidocs](https://apidocs.hackforums.net/)

Common write flows the agent can help describe:

- Reply to a thread: [hfapidocs](https://apidocs.hackforums.net/)
 - Use `/write` or `/write/posts` with `asks.posts._tid` and `asks.posts._message`.
- Create a thread: [hfapidocs](https://apidocs.hackforums.net/)
 - Use `POST /write/threads` with appropriate inputs (e.g., forum id, subject, message) according to the `asks` schema.
- Manage Bytes: [hfapidocs](https://apidocs.hackforums.net/)
 - `POST /write/bytes` to send Bytes (`_to_uid`, `_amount`).
 - `POST /write/bytes/deposit` to deposit to vault (minimum 100).
 - `POST /write/bytes/withdraw` to withdraw from vault (minimum 100).
 - `POST /write/bytes/bump` to bump a thread via Stanley Byte Bump.

[hfapidocs](https://apidocs.hackforums.net/)

The agent should remind users that writes must pass validation (correct inputs, scopes, etc.).

[hfapidocs](https://apidocs.hackforums.net/)

Errors, Limits, and Revocation

The agent should be able to reason about error handling and give best-practice advice.

- 401 Unauthorized – usually due to missing or invalid access token; ensure Bearer token is present and not expired or revoked. [hfapidocs](https://apidocs.hackforums.net/)
- Rate limiting – responses may include `x-rate-limit-remaining` header that callers should monitor to avoid throttling. [hfapidocs](https://apidocs.hackforums.net/)
- Validation errors – write requests may fail if required input fields or scopes are missing. [hfapidocs](https://apidocs.hackforums.net/)
- Revocation – users can revoke app access from their Authorized Apps page; apps should detect invalid tokens and trigger re-authorization. [hfapidocs](https://apidocs.hackforums.net/)

Agent Capabilities & Use Cases

In conversations, the agent can:

- Explain how to register a **Hack Forums** developer app and which fields to fill out for specific use cases (e.g., bots, dashboards, trading tools). [hfapidocs](https://apidocs.hackforums.net/)
- Help choose minimal scopes for use cases like profile dashboards, post analysis, Bytes management tools, or contract trackers. [hfapidocs](https://apidocs.hackforums.net/)
- Draft example HTTP requests (headers, JSON bodies) for `/read` and `/write` endpoints, including nested `asks` structures. [hfapidocs](https://apidocs.hackforums.net/)
- Debug common integration issues (wrong endpoint, missing `\_` inputs, scope mismatches, 401s, rate limiting). [hfapidocs](https://apidocs.hackforums.net/)
- Suggest secure architecture patterns: backend-only token storage, rotating secrets, handling revocation flows. [hfapidocs](https://apidocs.hackforums.net/)
- Point users to community resources such as the PHP example by Xerotic and the Python integration by ENKI on GitHub. [hfapidocs](https://apidocs.hackforums.net/)

Limitations & Safety Notes

- The agent cannot obtain or manage real tokens or secrets; it only explains how to use them. [hfapidocs](https://apidocs.hackforums.net/)
- The agent should never suggest hard-coding `client\_secret` or access tokens into public repos or client-side code. [hfapidocs](https://apidocs.hackforums.net/)
- The documentation referenced is unofficial; behavior may change and users should verify against the latest official materials if available. [hfapidocs](https://apidocs.hackforums.net/)

Vectura - October 21, 2025, 3:18 pm

Then we say, thank you [Vectura](#) for bringing to light how difficult the API is to understand; but despite that, thank you Vectura for still managing to put everything together to create a working Hackforums dash ?

But really bud, HQ thread, thanks!

Oh_Well - October 21, 2025, 3:20 pm

(Oct 21, 2025 03:18 PM)Vectura Wrote: [\[CITE\]](#) Then we say, thank you [Vectura](#) for bringing to light how difficult the API is to understand; but despite that, thank you Vectura for still managing to put

everything together to create a working Hackforums dash ?

But really bud, HQ thread, thanks!

Thank you [Vectura](#) for bringing to light how difficult the API is to understand ??

Vectura - October 21, 2025, 3:21 pm

(Oct 21, 2025 03:20 PM)Oh_Well Wrote: [\[CITE\]](#) Thank you [Vectura](#) for bringing to light how difficult the API is to understand ??

Hahahahaha, no, that API is really hard to understand for a layman. The error handling built into it is really rubbish.

genesai - October 21, 2025, 3:33 pm

Excuse the ignorance for a second but what does HF API allow us to do? What do we need to integrated with HF for?

I mean this in no bad way, just genuine curiosity

Vectura - October 21, 2025, 3:44 pm

(Oct 21, 2025 03:20 PM)Oh_Well Wrote: [\[CITE\]](#) Thank you [Vectura](#) for bringing to light how difficult the API is to understand ??

Hahahahaha, no, that API is really hard to understand for a layman. The error handling built into it is really rubbish.

PhillyStyle - October 21, 2025, 4:14 pm

I was thinking of making an program that lists the most recent threads/replies in selected forums. So that I don't have to click on each forum to see what is at the top of the list. Does that sound like something that could be done with the API?

Never mind, I don't have a Redirect URI.

Oh_Well - October 21, 2025, 11:48 pm

(Oct 21, 2025 03:33 PM)genesai Wrote: [\[CITE\]](#) Excuse the ignorance for a second but what does HF API allow us to do? What do we need to integrated with HF for?

I mean this in no bad way, just genuine curiosity

You can do lot's of stuff!

Take a look at the dashboard [Vectura](#) created and you'll get a taste!

<https://hackforums.net/showthread.php?tid=6310504>

You can also run analytics, send automated reminders, lookup users, transfer bytes, manage contracts etc.

CL1B - October 23, 2025, 10:45 am

Woah, this is quite nicely done! GG!

[Omniscient](#) what you think?

Novo - October 23, 2025, 10:47 am

YOU ARE THE MAN Oh_Well!

xadamxk - October 26, 2025, 6:37 pm

Did you reference these docs at all?

<https://github.com/xadamxk/hf-api-specification>

I'd be curious to know if the API specs have changed since this was written.

Terry A. Davis - November 26, 2025, 5:07 pm

I appreciate immensely that Swagger docs were provided.

Since learning about it, and using it, I am a massive fan for documentation, not like I'll ever write an Agent for HF, but nice to see that it is documented in a nice way.

Oh_Well - March 10, 2026, 3:29 am

Website is back online. It was down for a few feeks and I didn't realize it.

Omniscient - March 22, 2026, 12:46 pm

Doesn't everyone Vibe Code now the HF API? I would think something for a skills.md file would be better.

CL1B - March 23, 2026, 2:07 am

(Mar 22, 2026 12:46 PM)Omniscient Wrote: [\[CITE\]](#) Doesn't everyone Vibe Code now the HF API? I would think something for a skills.md file would be better.

Whether it's agentic or human coding, nicely written documentations go a long way.

Omniscient - March 23, 2026, 10:23 am

Unbelievable. I have such bad luck sometimes.

I spoke with [Oh_Well](#) and got permission to host this on an HF subdomain. I'm thinking to use the Wiki server. It's just some static pages, no big deal.

Unfortunately, I didn't setup the Wiki server and it's all very basic and manual commands (no Control Panel). That's fine, I go to edit the confs for Nginx, seemingly goes well. I try to install a cert (Cerbot w/Letsencrypt) and I get an error. It's older Debian server so I go start an apt update. That doesn't go well. Now I'm locked out of the server, pretty sure if I restart it that it won't come up. Wiki online for now and I've done a DB backup and luckily I had a connection open with SFTP and I'm trying to DL a full site backup.

What a waste of my time. This will likely kill my entire Monday. I have to get this backup done the reinstall the OS and build the server from scratch again (with Control Panel this time) and setup both the wiki and the apidocs.

Insane.

7 - March 23, 2026, 10:34 am

But this is very useful thread for what I am looking into also! Thanks

Do you not use origin certificates?

Could we also get a MD or json link for AI training? Rather than scraping the page

Oh_Well - March 23, 2026, 10:42 am

(Mar 23, 2026 10:34 AM)7 Wrote: [\[CITE\]](#) Could we also get a MD or json link for AI training? Rather than scraping the page

Here you go!

skills.md

Code HF API v2 Skills

=====

This file describes how an AI agent can understand and work with the **Hack Forums API v2** to help users design, debug, or reason about integrations. [hfapidocs](https://hfapidocs.site/)

Core Concept

- HF API v2 is an OAuth-based HTTP API that exposes read and write operations for Hack Forums resources (users, posts, threads, bytes, contracts, etc.) via a unified `/read` and `/write` model plus helper endpoints. [hfapidocs](https://hfapidocs.site/)

Base Details

- Base URL: `https://hackforums.net/api/v2`. [hfapidocs](https://hfapidocs.site/)
- Main methods: `POST` with JSON bodies using an `asks` object to describe requested resources and fields. [hfapidocs](https://hfapidocs.site/)
- Authentication: `Authorization: Bearer ACCESS_TOKEN` header with OAuth 2.0 access tokens. [hfapidocs](https://hfapidocs.site/)

OAuth & App Setup

The agent should know how to describe and reason about the OAuth flow, but not execute it itself.

- Developer apps are created in the HF Developer Portal (`usercp.php?action=apideveloper`) where you define name, description, redirect URI, details, and requested permissions (scopes). [hfapidocs](https://hfapidocs.site/)
- After approval, you receive a `client_id` and `client_secret` used in the OAuth authorization code flow. [hfapidocs](https://hfapidocs.site/)
- OAuth steps: [hfapidocs](https://hfapidocs.site/)
 1. Redirect user to HF authorize page with `client_id`, `response_type=code`, optional `state`.
 2. User grants scopes; HF redirects back to your redirect URI with `code` and `state` (if supplied).
 3. Exchange `code` for an access token via `POST https://hackforums.net/api/v2/authorize` using form-encoded body (`grant_type=authorization_code`, `client_id`, `client_secret`, `code`).
 4. Store the `access_token` securely and send it as a Bearer token on API calls.

Security constraints the agent should remind users of:

- Never put `client_secret` or access tokens in front-end or public code. [hfapidocs](https://hfapidocs.site/)
- Rotate tokens if compromised and handle revocation and 401 responses gracefully. [hfapidocs](https://hfapidocs.site/)

Scopes & Permissions

The agent should help users choose minimal scopes and understand what each scope unlocks. [hfapidocs](https://hfapidocs.site/)

Main scopes (during beta):

- Basic Info – public profile of the authorized user (e.g., `uid`, `username`, `usergroup`). [hfapidocs](https://hfapidocs.site/)
- Advanced Info – private profile details of the authorized user (e.g., `unreadpms`, `invisible`,

`totalpms`). [hfapids](https://hfapids.site/)

- Posts – access to forums, threads, posts; includes optional write permissions.

[hfapids](https://hfapids.site/)

- Users – read public info of other members (`uid`, `username`, `avatar`, etc.).

[hfapids](https://hfapids.site/)

- Bytes – read Byte logs plus optional write actions (transfers, deposits, withdrawals, bumps).

[hfapids](https://hfapids.site/)

- Contracts – access contracts, disputes, and b-ratings. [hfapids](https://hfapids.site/)

Scope changes:

- Increasing permissions requires users to re-authorize the application.

[hfapids](https://hfapids.site/)

Reading Data (`/read`)

General pattern:

- Endpoint: `POST /read` or resource-specific helpers like `/read/posts`, `/read/me`, etc.

[hfapids](https://hfapids.site/)

- Body structure uses an `asks` object; inputs to resources are prefixed with `\_` (e.g., `\_pid`, `\_tid`).

[hfapids](https://hfapids.site/)

- Fields set to `true` are requested back in the response. [hfapids](https://hfapids.site/)

Example patterns the agent should be able to describe or generate:

- Fetch current user data plus a specific thread: [hfapids](https://hfapids.site/)

- Use `/read` with `asks.me` for user fields (e.g., `uid`, `username`, `usergroup`) and `asks.threads` with `\_tid` and nested fields like `firstpost.message` and author info.

- Fetch posts by `pid`: [hfapids](https://hfapids.site/)

- Use `/read/posts` with `asks.posts.\_pid` as a list and booleans for desired post fields (e.g., `pid`, `message`, `subject`, `dateline`, etc.).

Key read endpoints the agent should recognize: [hfapids](https://hfapids.site/)

- `POST /read` – generic nested reads.

- `POST /read/me` – current authorized user profile.

- `POST /read/users` – public info of other users.

- `POST /read/forums` – forum data.

- `POST /read/threads` – threads by `tid`.

- `POST /read/posts` – posts by `pid`.

- `POST /read/bytes` – Byte logs.

- `POST /read/contracts` – contracts data.

- `POST /read/disputes` – disputes data.

- `POST /read/bratings` – b-ratings.

- `POST /read/sigmarket/market` and `POST /read/sigmarket/order` – signature market resources.

[hfapids](https://hfapids.site/)

Writing Data (`/write`)

General pattern:

- Endpoint: `POST /write` or resource-specific helpers such as `/write/posts`, `/write/bytes`,

`/write/bytes/deposit` , /write/bytes/withdraw` , /write/bytes/bump` . [hfapidocs](https://hfapidocs.site/)`

- Requires corresponding write scope for the resource (e.g., Posts, Bytes).

[hfapidocs](https://hfapidocs.site/)

- Body uses an `asks` object with underscore-prefixed input fields (e.g., _tid` , _message` , _amount` , _to_uid`). [hfapidocs](https://hfapidocs.site/)`

Common write flows the agent can help describe:

- Reply to a thread: [hfapidocs](https://hfapidocs.site/)
 - Use `/write` or /write/posts` with asks.posts._tid` and asks.posts._message` .`
- Create a thread: [hfapidocs](https://hfapidocs.site/)
 - Use `POST /write/threads` with appropriate inputs (e.g., forum id, subject, message) according to the asks` schema.`
- Manage Bytes: [hfapidocs](https://hfapidocs.site/)
 - `POST /write/bytes` to send Bytes (_to_uid` , _amount`).`
 - `POST /write/bytes/deposit` to deposit to vault (minimum 100).`
 - `POST /write/bytes/withdraw` to withdraw from vault (minimum 100).`
 - `POST /write/bytes/bump` to bump a thread via Stanley Byte Bump.`

[hfapidocs](https://hfapidocs.site/)

The agent should remind users that writes must pass validation (correct inputs, scopes, etc.).

[hfapidocs](https://hfapidocs.site/)

Errors, Limits, and Revocation

The agent should be able to reason about error handling and give best-practice advice.

- 401 Unauthorized – usually due to missing or invalid access token; ensure Bearer token is present and not expired or revoked. [hfapidocs](https://hfapidocs.site/)
- Rate limiting – responses may include `x-rate-limit-remaining` header that callers should monitor to avoid throttling. [hfapidocs](https://hfapidocs.site/)`
- Validation errors – write requests may fail if required input fields or scopes are missing.

[hfapidocs](https://hfapidocs.site/)

- Revocation – users can revoke app access from their Authorized Apps page; apps should detect invalid tokens and trigger re-authorization. [hfapidocs](https://hfapidocs.site/)

Agent Capabilities & Use Cases

In conversations, the agent can:

- Explain how to register a **Hack Forums** developer app and which fields to fill out for specific use cases (e.g., bots, dashboards, trading tools). [hfapidocs](https://hfapidocs.site/)
- Help choose minimal scopes for use cases like profile dashboards, post analysis, Bytes management tools, or contract trackers. [hfapidocs](https://hfapidocs.site/)
- Draft example HTTP requests (headers, JSON bodies) for `/read` and /write` endpoints, including nested asks` structures. [hfapidocs](https://hfapidocs.site/)`
- Debug common integration issues (wrong endpoint, missing `_` inputs, scope mismatches, 401s, rate limiting). [hfapidocs](https://hfapidocs.site/)`
- Suggest secure architecture patterns: backend-only token storage, rotating secrets, handling revocation flows. [hfapidocs](https://hfapidocs.site/)
- Point users to community resources such as the PHP example by Xerotic and the Python integration by ENKI on GitHub. [hfapidocs](https://hfapidocs.site/)

Limitations & Safety Notes

- The agent cannot obtain or manage real tokens or secrets; it only explains how to use them. [hfapdocs](https://hfapdocs.site/)
- The agent should never suggest hard-coding `client\_secret` or access tokens into public repos or client-side code. [hfapdocs](https://hfapdocs.site/)
- The documentation referenced is unofficial; behavior may change and users should verify against the latest official materials if available. [hfapdocs](https://hfapdocs.site/)

7 - March 23, 2026, 10:44 am

(Mar 23, 2026 10:42 AM)Oh_Well Wrote: [\[CITE\]](#) Here you go!

skills.md

Code HF API v2 Skills

=====

This file describes how an AI agent can understand and work with the **Hack Forums API v2** to help users design, debug, or reason about integrations. [hfapdocs](https://hfapdocs.site/)

Core Concept

- HF API v2 is an OAuth-based HTTP API that exposes read and write operations for Hack Forums resources (users, posts, threads, bytes, contracts, etc.) via a unified `/read` and `/write` model plus helper endpoints. [hfapdocs](https://hfapdocs.site/)

Base Details

- Base URL: `https://hackforums.net/api/v2`. [hfapdocs](https://hfapdocs.site/)
- Main methods: `POST` with JSON bodies using an `asks` object to describe requested resources and fields. [hfapdocs](https://hfapdocs.site/)
- Authentication: `Authorization: Bearer ACCESS\_TOKEN` header with OAuth 2.0 access tokens. [hfapdocs](https://hfapdocs.site/)

OAuth & App Setup

The agent should know how to describe and reason about the OAuth flow, but not execute it itself.

- Developer apps are created in the HF Developer Portal (`usercp.php?action=apideveloper`) where you define name, description, redirect URI, details, and requested permissions (scopes). [hfapdocs](https://hfapdocs.site/)
- After approval, you receive a `client\_id` and `client\_secret` used in the OAuth authorization code flow. [hfapdocs](https://hfapdocs.site/)
- OAuth steps: [hfapdocs](https://hfapdocs.site/)
 1. Redirect user to HF authorize page with `client\_id`, `response\_type=code`, optional `state`.
 2. User grants scopes; HF redirects back to your redirect URI with `code` and `state` (if supplied).
 3. Exchange `code` for an access token via `POST https://hackforums.net/api/v2/authorize` using form-encoded body (`grant\_type=authorization\_code`, `client\_id`, `client\_secret`, `code`).
 4. Store the `access\_token` securely and send it as a Bearer token on API calls.

Security constraints the agent should remind users of:

- Never put `client\_secret` or access tokens in front-end or public code.

[hfapidoes](https://hfapidoes.site/)

- Rotate tokens if compromised and handle revocation and 401 responses gracefully.

[hfapidoes](https://hfapidoes.site/)

Scopes & Permissions

The agent should help users choose minimal scopes and understand what each scope unlocks.

[hfapidoes](https://hfapidoes.site/)

Main scopes (during beta):

- Basic Info – public profile of the authorized user (e.g., `uid`, `username`, `usergroup`).

[hfapidoes](https://hfapidoes.site/)

- Advanced Info – private profile details of the authorized user (e.g., `unreadpms`, `invisible`, `totalpms`).

[hfapidoes](https://hfapidoes.site/)

- Posts – access to forums, threads, posts; includes optional write permissions.

[hfapidoes](https://hfapidoes.site/)

- Users – read public info of other members (`uid`, `username`, `avatar`, etc.).

[hfapidoes](https://hfapidoes.site/)

- Bytes – read Byte logs plus optional write actions (transfers, deposits, withdrawals, bumps).

[hfapidoes](https://hfapidoes.site/)

- Contracts – access contracts, disputes, and b-ratings. [hfapidoes](https://hfapidoes.site/)

Scope changes:

- Increasing permissions requires users to re-authorize the application.

[hfapidoes](https://hfapidoes.site/)

Reading Data (`/read`)

General pattern:

- Endpoint: `POST /read` or resource-specific helpers like `/read/posts`, `/read/me`, etc.

[hfapidoes](https://hfapidoes.site/)

- Body structure uses an `asks` object; inputs to resources are prefixed with `\_` (e.g., `\_pid`, `\_tid`).

[hfapidoes](https://hfapidoes.site/)

- Fields set to `true` are requested back in the response. [hfapidoes](https://hfapidoes.site/)

Example patterns the agent should be able to describe or generate:

- Fetch current user data plus a specific thread: [hfapidoes](https://hfapidoes.site/)

- Use `/read` with `asks.me` for user fields (e.g., `uid`, `username`, `usergroup`) and `asks.threads` with `\_tid` and nested fields like `firstpost.message` and author info.

- Fetch posts by `pid`: [hfapidoes](https://hfapidoes.site/)

- Use `/read/posts` with `asks.posts.\_pid` as a list and booleans for desired post fields (e.g., `pid`, `message`, `subject`, `dateline`, etc.).

Key read endpoints the agent should recognize: [hfapidoes](https://hfapidoes.site/)

- `POST /read` – generic nested reads.
- `POST /read/me` – current authorized user profile.
- `POST /read/users` – public info of other users.
- `POST /read/forums` – forum data.

- `POST /read/threads`` – threads by `tid``.
- `POST /read/posts`` – posts by `pid``.
- `POST /read/bytes`` – Byte logs.
- `POST /read/contracts`` – contracts data.
- `POST /read/disputes`` – disputes data.
- `POST /read/bratings`` – b-ratings.
- `POST /read/sigmarket/market`` and `POST /read/sigmarket/order`` – signature market resources.

[hfapidocs](https://hfapidocs.site/)

Writing Data (`/write``)

General pattern:

- Endpoint: `POST /write`` or resource-specific helpers such as `/write/posts``, `/write/bytes``, `/write/bytes/deposit``, `/write/bytes/withdraw``, `/write/bytes/bump``. [hfapidocs](https://hfapidocs.site/)
- Requires corresponding write scope for the resource (e.g., Posts, Bytes).

[hfapidocs](https://hfapidocs.site/)

- Body uses an `asks`` object with underscore-prefixed input fields (e.g., `_tid``, `_message``, `_amount``, `_to_uid``). [hfapidocs](https://hfapidocs.site/)

Common write flows the agent can help describe:

- Reply to a thread: [hfapidocs](https://hfapidocs.site/)
 - Use `/write`` or `/write/posts`` with `asks.posts._tid`` and `asks.posts._message``.
- Create a thread: [hfapidocs](https://hfapidocs.site/)
 - Use `POST /write/threads`` with appropriate inputs (e.g., forum id, subject, message) according to the `asks`` schema.
- Manage Bytes: [hfapidocs](https://hfapidocs.site/)
 - `POST /write/bytes`` to send Bytes (`_to_uid``, `_amount``).
 - `POST /write/bytes/deposit`` to deposit to vault (minimum 100).
 - `POST /write/bytes/withdraw`` to withdraw from vault (minimum 100).
 - `POST /write/bytes/bump`` to bump a thread via Stanley Byte Bump.

[hfapidocs](https://hfapidocs.site/)

The agent should remind users that writes must pass validation (correct inputs, scopes, etc.).

[hfapidocs](https://hfapidocs.site/)

Errors, Limits, and Revocation

The agent should be able to reason about error handling and give best-practice advice.

- 401 Unauthorized – usually due to missing or invalid access token; ensure Bearer token is present and not expired or revoked. [hfapidocs](https://hfapidocs.site/)
- Rate limiting – responses may include `x-rate-limit-remaining`` header that callers should monitor to avoid throttling. [hfapidocs](https://hfapidocs.site/)
- Validation errors – write requests may fail if required input fields or scopes are missing.

[hfapidocs](https://hfapidocs.site/)

- Revocation – users can revoke app access from their Authorized Apps page; apps should detect invalid tokens and trigger re-authorization. [hfapidocs](https://hfapidocs.site/)

Agent Capabilities & Use Cases

In conversations, the agent can:

- Explain how to register a **Hack Forums** developer app and which fields to fill out for specific use cases (e.g., bots, dashboards, trading tools). [hfapidocs](https://hfapidocs.site/)
- Help choose minimal scopes for use cases like profile dashboards, post analysis, Bytes management tools, or contract trackers. [hfapidocs](https://hfapidocs.site/)
- Draft example HTTP requests (headers, JSON bodies) for `/read` and `/write` endpoints, including nested `asks` structures. [hfapidocs](https://hfapidocs.site/)
- Debug common integration issues (wrong endpoint, missing `_`` inputs, scope mismatches, 401s, rate limiting). [hfapidocs](https://hfapidocs.site/)
- Suggest secure architecture patterns: backend-only token storage, rotating secrets, handling revocation flows. [hfapidocs](https://hfapidocs.site/)
- Point users to community resources such as the PHP example by Xerotic and the Python integration by ENKI on GitHub. [hfapidocs](https://hfapidocs.site/)

Limitations & Safety Notes

- The agent cannot obtain or manage real tokens or secrets; it only explains how to use them. [hfapidocs](https://hfapidocs.site/)
- The agent should never suggest hard-coding `client_secret` or access tokens into public repos or client-side code. [hfapidocs](https://hfapidocs.site/)
- The documentation referenced is unofficial; behavior may change and users should verify against the latest official materials if available. [hfapidocs](https://hfapidocs.site/)

Thanks mate! Appreciated.

I was thinking about making a markdown scraper for API's globally.

Considering the way tech is going. It's much easier than rescraping the api docs

Oh_Well - March 23, 2026, 10:49 am

(Mar 23, 2026 10:23 AM)Omniscient Wrote: [\[CITE\]](#) Unbelievable. I have such bad luck sometimes.

I spoke with [Oh_Well](#) and got permission to host this on an HF subdomain. I'm thinking to use the Wiki server. It's just some static pages, no big deal.

Unfortunately, I didn't setup the Wiki server and it's all very basic and manual commands (no Control Panel). That's fine, I go to edit the confs for Nginx, seemingly goes well. I try to install a cert (Cerot w/Letsencrypt) and I get an error. It's older Debian server so I go start an apt update. That doesn't go well. Now I'm locked out of the server, pretty sure if I restart it that it won't come up. Wiki online for now and I've done a DB backup and luckily I had a connection open with SFTP and I'm trying to DL a full site backup.

What a waste of my time. This will likely kill my entire Monday. I have to get this backup done the reinstall the OS and build the server from scratch again (with Control Panel this time) and setup both the wiki and the apidocs.

Insane.

That's bad luck, sorry for triggering that!!

(Mar 23, 2026 10:44 AM)7 Wrote: [\[CITE\]](#) Thanks mate! Appreciated.

I was thinking about making a markdown scraper for API's globally.

Considering the way tech is going. It's much easier than rescraping the api docs
Have you considered using moltbook to exchange docs/API info?

Omniscient - March 23, 2026, 10:50 am

(Mar 23, 2026 10:49 AM)Oh_Well Wrote: [\[CITE\]](#) That's bad luck, sorry for triggering that!!

Have you considered using moltbook to exchange docs/API info?
I haven't yet setup an AI agent.

Oh_Well - March 24, 2026, 7:08 am

Omni got it working. So the HF API Documentation page is now under

apidocs.hackforums.net

Thank you [Omniscient](#) for taking the time to host this, and I hope the HF community uses it to its fullest potential!

Omniscient - March 24, 2026, 9:59 am

<https://apidocs.hackforums.net/SKILLS.md>

Added the SKILLS page.

btw, apparently it should be capitalized.

The swagger docs are going to be moved to Swagger.HackForums.net once [xadamxk](#) tells me the site url. I overwrote the apidocs cname and forgot to save the original url to do the swap. Oops.

CL1B - March 24, 2026, 11:14 am

(Mar 24, 2026 07:08 AM)Oh_Well Wrote: [\[CITE\]](#) Omni got it working. So the HF API Documentation page is now under

apidocs.hackforums.net

Thank you [Omniscient](#) for taking the time to host this, and I hope the HF community uses it to its fullest potential!

Nice, I was looking for this just now; I saw the old domain was down, checks out.

xadamxk - March 24, 2026, 1:34 pm

(Mar 24, 2026 09:59 AM)Omniscient Wrote: [\[CITE\]](#) <https://apidocs.hackforums.net/SKILLS.md>

Added the SKILLS page.

btw, apparently it should be capitalized.

The swagger docs are going to be moved to Swagger.HackForums.net once [xadamxk](#) tells me the site url. I overwrote the apidocs cname and forgot to save the original url to do the swap. Oops.

Running some errands atm and am not signed into GitHub on my phone. I'll double check when I get home and DM you.

Alright got everything updated on my end. <http://swagger.hackforums.net/>

xadamxk - May 23, 2026, 11:32 am

I'm trying to READ threads from a specific user in a specific subforum.

I have this, but the _fid isn't applying for some reason. Any idea why?

Also, how does pagination / sort order work? I can't find shit when it comes to pagination / offsets

haha

For example:

```
Code {  
  "asks": {  
    "threads": {  
      "_fid": 2,  
      "_uid": 1,  
      "tid": true,  
      "uid": true,  
      "subject": true,  
      "dateline": true,  
      "fid": true  
    }  
  }  
}
```

EDIT: Turns out the threads search parameters only support these and not fid :(

```
Code '_uid' => [59852261], // ...or search by UID and optional _page/_perpage parameters  
  '_page' => [123],  
  '_perpage' => [123],
```

Here are the actual "docs" if anyone else needs them:

<https://hackforums.net/showthread.php?tid=6083782>

Au Jus Demon - May 23, 2026, 11:49 am

(May 23, 2026 11:32 AM)xadamxk Wrote: [\[CITE\]](#) I'm trying to READ threads from a specific user in a specific subforum.

I have this, but the _fid isn't applying for some reason. Any idea why?

Also, how does pagination / sort order work? I can't find shit when it comes to pagination / offsets
haha

For example:

```
Code {  
  "asks": {  
    "threads": {  
      "_fid": 2,  
      "_uid": 1,  
      "tid": true,  
      "uid": true,  
      "subject": true,  
      "dateline": true,  
      "fid": true  
    }  
  }  
}
```

EDIT: Turns out the threads search parameters only support these and not fid :(

```
Code '_uid' => [59852261], // ...or search by UID and optional _page/_perpage parameters  
  '_page' => [123],  
  '_perpage' => [123],
```

Here are the actual "docs" if anyone else needs them:

<https://hackforums.net/showthread.php?tid=6083782>

Not 100% sure this is the intended way but what worked for me was fetching by `_uid` and filtering on fid client side after

for pagination: `_page` starts at 1, `_perpage` caps at 30, no offset. when you get back fewer rows than `_perpage` you're on the last page. `_uid` sorts newest activity first

also worth noting `_uid` is OP-only, won't return threads the user replied to but didn't create. need `posts._uid` for that

`_fid` and `_uid` just can't be combined on the same threads call, `_uid` overrides it