

How to Connect DeepSeek to Janitor AI Using API

TechRounder PDF Edition

Live article: <https://www.techrounder.com/how-to/how-to-connect-deepseek-to-janitor-ai-using-api/>

By Vipin PG | Published April 23, 2026 | Updated April 23, 2026 | Format: Guide | 8 min read

Quick answer

This article explains how to enhance Janitor AI by connecting it to a DeepSeek API key using a custom proxy configuration. By following a specific setup of the API endpoint and model names, users can bypass default routes to achieve more reliable and customizable chat experiences.

Key points

- Core Integration: Janitor AI can connect to DeepSeek via the "Custom Proxy" mode because DeepSeek uses an OpenAI-compatible API.
- Required Values: To function correctly, users must use the endpoint 'https://api.deepseek.com/chat/completions' and the model names 'deepseek-chat' (for general roleplay) or 'deepseek-reasoner' (for complex reasoning).
- Critical Setup Steps: Settings should be configured from within an active character chat, and the page must be refreshed after saving to ensure the configuration takes effect.
- Prerequisites: Users must have a valid DeepSeek API key and a positive account balance to avoid "402 Insufficient Balance" errors.
- Troubleshooting Tips: 401 errors typically indicate an incorrect API key, 404 errors suggest an incorrect model name, and "Failed to fetch" errors often require a page refresh or a check of the DeepSeek status page.

Janitor AI becomes much more usable once you stop relying on whatever default route happens to be available and connect it to a model you control. DeepSeek is a strong fit for that because it exposes an OpenAI-compatible API, which means Janitor AI can talk to it through custom proxy fields instead of needing a special native integration.

The part that trips people up is not the API key itself. It is the mismatch between DeepSeek's documented API format and Janitor AI's UI labels. Most failed setups come down to one of three things: the wrong endpoint, the wrong model name, or saving the settings without refreshing the chat afterward. This guide uses the current working pattern and keeps those failure points visible from the start.

What You Will Build / Achieve

By the end of this setup, Janitor AI will send chat requests through your own DeepSeek API key instead of the default route. You will have a working custom proxy configuration using either 'deepseek-chat' or 'deepseek-reasoner', and you will verify it by validating the key inside Janitor AI and sending a live test prompt that returns a DeepSeek-generated reply.

Prerequisites

- A Janitor AI account with access to a character chat where API settings can be opened from the conversation screen.
- A DeepSeek platform account with a generated API key from DeepSeek API keys .

- Available DeepSeek account balance if you plan to use the paid official API directly. DeepSeek returns '402' when the balance is empty.
- A current desktop or mobile browser. The screens can look slightly different, but the fields and values are the same.
- A stable internet connection. If DeepSeek is having an incident, Janitor AI can look misconfigured even when your values are correct.
- A few minutes to test both model modes. 'deepseek-chat' is the safer starting point for general roleplay and chat. 'deepseek-reasoner' is better when you want slower, more deliberate reasoning.
- If DeepSeek has been unstable for you lately, keep this DeepSeek error guide open in another tab for fallback ideas.

This guide covers the direct official DeepSeek API connection inside Janitor AI. It does not cover OpenRouter, Chutes, or self-hosted DeepSeek replacements. If you later want to move away from hosted usage, TechRounder's self-hosted LLM guide is the logical next step.

Prepare the DeepSeek API Side

Step 1: Create a DeepSeek API key

Sign in to the DeepSeek platform and generate a new key from the API key page.

The DeepSeek API is documented as OpenAI-compatible in the DeepSeek API docs, which is the reason Janitor AI can use it through custom proxy settings.

Once the key is created, copy it immediately and store it somewhere safe. You should treat it like a password because Janitor AI will use it to bill requests against your DeepSeek account.

The expected result is that you have a full key string ready to paste, not a partial copy and not a screenshot of the key.

Step 2: Confirm you have enough API balance

Open your DeepSeek billing view and make sure the account has usable balance before you test the connection.

If the key is valid but the balance is empty, the request can still fail. DeepSeek documents this as a '402 - Insufficient Balance' condition on its official error codes page.

The expected result is that your account shows available funds or credits before you move to Janitor AI.

Step 3: Choose the model you want to start with

Pick one model before you open Janitor AI so you are not guessing in the middle of setup.

Model name to enter: 'deepseek-chat' | Best use: General chat and roleplay | What to expect: Faster, more natural default choice for most beginners

Model name to enter: 'deepseek-reasoner' | Best use: Reasoning-heavy prompts | What to expect: More deliberate output, usually with higher latency

DeepSeek's current documentation maps both names to DeepSeek-V3.2, which is why older articles that tell you to paste older version strings often break.

The expected result is that you have chosen one exact model name and copied it correctly.

Configure Janitor AI

Step 1: Open a character chat first

Open Janitor AI and enter the specific character chat you want to use for testing.

This matters because the working setup path is tied to the active chat screen in current Janitor AI web UI. Community walkthroughs repeatedly show that users get less predictable results when they try to start from a generic settings area.

The expected result is that you are inside a live chat view, not only on the homepage or character list.

Step 2: Open the API settings from the chat screen

Use the menu or status control inside the chat to open Janitor AI's API settings.

Depending on the current UI, this may appear as a status label such as "Using janitor" or "Using proxy," or inside a menu opened from the chat header. If you do not see the API settings immediately, stay inside the chat and check the top-right controls first.

The expected result is that the API settings panel is open and you can switch providers or proxy options.

Step 3: Switch to the custom proxy mode

Change the provider or route to the custom proxy option.

In most current setups, this means selecting 'Proxy' and then choosing 'Custom'. Janitor AI uses this path for OpenAI-compatible providers that are not built in as a native one-click option.

The expected result is that Janitor AI shows fields for a model name, an API key, and a proxy or API URL.

Step 4: Enter the working DeepSeek values

Paste the following values exactly as shown into the Janitor AI custom fields:

Model:

deepseek-chat

API Key:

YOUR_DEEPSEEK_API_KEY

Other API/Proxy URL:

https://api.deepseek.com/chat/completions

Replace 'YOUR_DEEPSEEK_API_KEY' with the exact key you generated in DeepSeek. Do not add quotes, spaces, or extra characters before or after it.

If you want the reasoning model instead, replace 'deepseek-chat' with 'deepseek-reasoner'. The endpoint stays the same.

The expected result is that all three fields are filled and the URL ends with '/chat/completions'.

Step 5: Save the settings before testing

Save the Janitor AI API settings once the fields are filled.

Quote: Warning: Do not keep editing the fields after you start testing. A surprising number of Janitor AI errors come from half-edited values, extra spaces, or switching models and URLs without saving the final combination first.

The expected result is that Janitor AI stores the values and returns you to the chat with the new route selected.

Step 6: Refresh the chat page

Refresh the current Janitor AI tab after saving the proxy configuration.

This is one of the least obvious but most useful steps. Community fixes for Janitor AI repeatedly point to a save-and-refresh cycle before the key check and first prompt behave normally.

The expected result is that the chat reloads with the custom DeepSeek settings still present.

Test the Connection

Step 1: Validate the API key inside Janitor AI

Use the built-in key check or validation control if Janitor AI shows one in the API settings panel.

If the validation succeeds, you should see a success state rather than a proxy error. If the page does nothing, reopen the API settings after the refresh and test again.

The expected result is a successful validation or at least no immediate authentication error.

Step 2: Send a minimal prompt

Start with a short test prompt rather than a long roleplay message.

Hello. Reply with the single word: connected

A short prompt removes most of the noise. If it fails, you know the problem is the connection itself, not your jailbreak, persona card, or token-heavy system prompt.

The expected result is a reply containing 'connected' or a close equivalent generated by DeepSeek.

Step 3: Switch models only after the first reply works

Once 'deepseek-chat' works, go back and test 'deepseek-reasoner' if you want a more deliberate model.

Changing only one variable at a time makes the failure point obvious. If the chat model works and the reasoner does not, you can focus on model compatibility or rate/latency issues instead of rechecking the whole setup.

The expected result is that both models can be tested independently with the same key and endpoint.

Verification

Step 1: Confirm the configuration survived a reload

Refresh Janitor AI one more time, reopen the API settings, and make sure the model, key slot, and endpoint are still set to your DeepSeek values.

The expected result is that the proxy configuration remains saved and the endpoint still ends with '/chat/completions'.

Step 2: Run a second prompt with a style check

Send a slightly more natural prompt to confirm you are not only getting a one-word reply.

Write two short sentences explaining what an API key does.

If the setup is working, Janitor AI should return a normal multi-sentence answer without throwing a proxy or network error.

The expected result is a clean response generated through DeepSeek, not a timeout screen, not a 401/404 proxy error, and not a generic fetch failure.

Step 3: Check service health if replies suddenly stop

If the exact same saved setup stops working after it already worked once, check the DeepSeek status page before changing your Janitor AI fields again.

This saves time because provider-side incidents can look exactly like a broken local configuration from inside Janitor AI.

The expected result is that you can separate a true misconfiguration from an upstream DeepSeek outage.

Troubleshooting

PROXY ERROR 401 or "Authentication Fails"

This nearly always means the API key is wrong, incomplete, expired, or copied from the wrong provider account.

Generate a fresh DeepSeek key, paste only the raw key value, save the settings, refresh the chat, and test again. If you are reusing a key from an older tab or screenshot, replace it with a newly copied one.

PROXY ERROR 404 or "model not found"

This usually happens when you paste a model slug from an older guide or from another provider such as OpenRouter.

For the official direct DeepSeek setup, use 'deepseek-chat' or 'deepseek-reasoner'. Do not paste older OpenRouter-style names or experimental release names unless you are intentionally routing through a different provider.

"Network error occurred, you may be rate limited or having connection issues: Failed to fetch"

This is often caused by one of two things: the wrong endpoint format or Janitor AI needing a refresh after the settings were saved.

Make sure your URL is exactly 'https://api.deepseek.com/chat/completions'. Then save the fields again, refresh Janitor AI, reopen the chat, and retry a short prompt. If it still fails, check whether DeepSeek has an active incident.

The key validates, but the bot still will not reply

A valid key does not guarantee the account has usable balance or that the upstream service is healthy.

Check your DeepSeek balance first. If billing is fine, check service health next. If both are normal, switch to 'deepseek-chat' for testing because it is usually the cleaner baseline. This is also a good point to review how LLMs shape chatbots if you are comparing model behavior rather than only connectivity.

The endpoint works with one guide but not another

Most of the confusion comes from mixing DeepSeek base URLs with Janitor AI's proxy URL field.

DeepSeek documents the base URL in its API docs, but Janitor AI custom proxy setups commonly work better when you paste the full chat-completions endpoint. Keep the model name separate from the URL, and do not add extra path fragments after '/chat/completions'.

Where to Go After This

Once the direct connection is stable, the next decisions are about cost, response style, and interface. Test both DeepSeek model modes on the same character, keep an eye on token usage, and decide whether Janitor AI is your long-term front end or only your first stop. If you want a broader local or desktop workflow after this, TechRounder's multi-LLM macOS apps article is a useful comparison point.

References

1. platform.deepseek.com - api_keys - https://platform.deepseek.com/api_keys
2. api-docs.deepseek.com - <https://api-docs.deepseek.com/>
3. api-docs.deepseek.com - quick_start / error_codes - https://api-docs.deepseek.com/quick_start/error_codes
4. status.deepseek.com - <https://status.deepseek.com/>