Discourse AI Customization Guide

67 messages

Generated on Fri May 23 2025 23:32:24 GMT-0400 (Eastern Daylight Time)

User

Tue May 20 2025 18:40:03 GMT-0400 (Eastern Daylight Time)

I'd like to understand better how to use Discourse AI more effectively. I've configured the OpenAI API Key, and used the available models to enable various functions, but newer models are clearly missing, notably 40. I used o1 as the model which generated Dall-E image, which worked, but I feel this will cost more than it needs to. I also would like to understand how to add my own code to expand the capabilities of AI agents in Discourse (notably creating AI Curators as partners of category moderators)

Assistant (*gpt-40*)

Tue May 20 2025 18:40:05 GMT-0400 (Eastern Daylight Time)

User is configuring Discourse AI features with their own OpenAI API key, aiming to expand AI agent capabilities such as creating AI Curators to assist category moderators.

Tue May 20 2025 18:40:09 GMT-0400 (Eastern Daylight Time)

You're taking a powerful approach by integrating AI into Discourse, especially with the goal of creating "AI Curators" as intelligent co-moderators. Here's a breakdown of how to improve your current setup and extend it:



Current State: Discourse AI with OpenAI API

You've:

- Installed Discourse with the Discourse AI plugin
- Configured an OpenAI API key
- Enabled available models (e.g. GPT-3.5-turbo, GPT-4, o1 for DALL-E)
- Noted that GPT-40 is not yet listed or usable via dropdown

Limitations and Immediate Enhancements

1. GPT-40 and Custom Model Access

As of now, Discourse AI plugin lags behind OpenAI's newest model support (like gpt-40). You can:

- Manually override the model in plugin settings or code.
- Check sitesettings.yml or relevant initializer files in discourse-ai for default model specs.
- You may also override per-function (e.g., summarizer vs. classifier).
- Edit YAML directly in plugin if dropdowns don't list all models:

```
vaml
 ai</em>openai<em>completions</em>model: gpt-4o
>
```

2. Cost Optimization

- o1 is probably pointing to dall-e-3, which is expensive.
- For image generation, you might route through a middleware service or queue to defer/restrict generation.
- Optionally, implement a credit/token system within Discourse to limit access by trust level, badge, or group.

X Expanding AI Functionality: Custom AI Curators

To create AI Curators as assistant moderators (essentially persistent, rule-aware agents), you'll need to:

Step 1: Hook into Categories

Each Discourse category can be extended via:

- Custom plugin using CategoryCustomFields
- Assigning AI Curators per category via a plugin schema like:

```
ruby
 category.custom<em>fields["ai</em>curator<em>prompt"] = "Moderate dis-
cussions on neuroscience with an eye for respectful disagreement and link
sharing."
<<p><<p><</p>
```

Step 2: Extend Plugin: Add Agent Workers

Inside the discourse-ai plugin (or a custom plugin that depends on it), you can:

- Define background jobs or long-running listeners using Jobs::Scheduled
- Add new class inheriting from DiscourseAi::AiBot::Base to define an agent

```
ruby
 module DiscourseAi
 module AiBot
 class CuratorAgent < Base
     def process(post)
>
       if relevant</em>to<em>category?(post)
>
        ai</em>summarize(post)
>
         ai<em>respond</em>if<em>needed(post)
>
>
       end
     end
>
    end
 end
 end
>
```

• Hook it into post creation:

```
ruby
 on(:post</em>created) do |post, opts, user|
 DiscourseAi::AiBot::CuratorAgent.new.process(post)
 end
<</p>
```

Step 3: Custom Prompts per Curator

You might allow defining curator personas:

- Add a custom admin interface (or use category description + tags)
- Feed persona prompt into system message of GPT call:

```
ruby
 system<em>prompt = SiteSetting.ai</em>curator_persona || "You are a
careful philosophical moderator..."
<<p><<p><</p>
```

Recommendations for Production-Ready Deployment

- Rate-limit AI responses per category to avoid overposting
- Log AI actions visibly with tags or special post formatting (e.g. [AI Curator Note])
- Fallbacks and monitoring: Add retry logic and log failures with

```
Discourse::Utils::Logster
```

Resources & Tools

- Discourse AI GitHub
- Discourse Plugin Developer's Guide
- For GPT-40: OpenAI API Documentation

☑Next Steps for You

Would you like me to help:

- Scaffold a Discourse plugin for AI Curators?
- Add GPT-40 model override safely?
- Integrate a basic credit/token limiter?

Let me know which path you want to focus on first, and I can generate the actual Ruby/plugin code to get you started.

Assistant (gpt-40)

Tue May 20 2025 18:40:16 GMT-0400 (Eastern Daylight Time)

User

Tue May 20 2025 18:46:26 GMT-0400 (Eastern Daylight Time)

Can I create a custom plugin that "hooks" into the Discourse AI plugin to add the newer models, without modifying the Discourse AI code? I'm thinking of the way Wordpress allows hooks to override functions, but without interfering with updating existing core or plugin code. Help me understand the available mechanisms in Discourse.

Tue May 20 2025 18:46:27 GMT-0400 (Eastern Daylight Time)

Yes, you can create a custom plugin that hooks into the Discourse AI plugin to extend its capabilities — including registering new models like gpt-40 — without modifying the core Discourse AI plugin directly. This is very much in line with Discourse's plugin architecture, which is inspired more by Ruby on Rails than WordPress, but offers flexible extension points.

Let me break this down:



🔧 1. Discourse Plugin Architecture Basics

Discourse plugins are Ruby code that integrate into the core via:

- Initializers (plugin.rb)
- Event hooks (on (:postcreated), etc.)
- Module overrides (via prepend, class eval, or Module#aliasmethod)
- Site settings and custom fields
- Registerable extensions (like AI models, embeddings, analyzers, bots)

The **Discourse AI plugin** is designed to support extensions, and it already registers different providers (like OpenAI, Anthropic, etc.) via internal registries.

2. Strategy to Add a New Model (like gpt-40) via

Custom Plugin

You can build a **separate plugin** that adds support for new models via:



A. Register a new model alias

You can hook into the DiscourseAi::Completions::OpenAiCompletions and add your own model mapping by patching it:

 $> \bigvee$ This overrides the model used when the plugin asks OpenAI for completions.

B. Add a new model to site settings

If the model list is constrained by allowed settings (SiteSetting.aiopenaicompletionsmodel choices), you can extend the setting options:

```
<h1&qt;pluqins/discourse-ai-extension/pluqin.rb&lt;/h1&qt;
  register</em>asset "javascripts/discourse/initializers/add<em>qpt4o</em>model.
  <h1&qt;assets/javascripts/discourse/initializers/add<em>qpt4o</em>model.js&
   lt;/h1>
   import { withPluginApi } from "discourse/lib/plugin-api";
  export default {
   name: "add-gpt4o-model",
   initialize() {
10
11
       withPluginApi("0.8", (api) => {
  >
        const siteSettings = api.container.lookup("site-settings:main");
  12
  >
         const modelSetting = siteSettings.ai<em>openai</em>completions<em>model;
13
   14 
         if (modelSetting && !modelSetting.choices.includes("gpt-4o"))
   {
15 
           modelSetting.choices.push("gpt-40");
16 
         }
17  });
18  },
19 | };
```

This allows the admin interface to show your model, assuming the backend can use it.

3. Avoiding Code Collisions

Discourse uses conventions like:

- after initialize blocks for safe plugin injection
- registerx style hooks (e.g. register asset, registermodel, register customfieldtype)

Avoiding core/plugin modifications ensures:

- You can safely update discourse-ai
- Your plugin can be toggled independently
- You maintain compatibility across Discourse upgrades

Use prepend or classeval for Ruby methods only if there's no public hook exposed — but these should be scoped carefully to avoid affecting unrelated functionality.

4. Recommended Practice for AI Plugins

The best approach would be:

- Create a plugin that defines its own subclass of the AI worker or tool, e.g.

 MyGpt4oSummarizer < ::DiscourseAi::Summarizer::Base
- Register it explicitly for certain categories or tags
- Provide fallback logic to the default summarizer otherwise

Summary: Hooking Into Discourse AI Safely

Goal	Method
Use new model (e.g., gpt-40)	prepend to override model-returning method
Add model to admin dropdown	Patch JavaScript via registerasset
Create new summarizer/chat tool	Subclass & register via initializer
Avoid breaking upgrades	Never modify discourse-ai directly; rely on plugin layering

Would you like a ready-to-deploy skeleton of this plugin (discourse-ai-extension) that adds gpt-40 and allows category-specific prompt customization for AI curators?

User

Tue May 20 2025 19:07:22 GMT-0400 (Eastern Daylight Time)

For additional AI models, I'd like to support Google's Gemini and Gemma models, the Groq.com models which include many open source models (served with high performance) and anthropic models as well as a full complement of the current OpenAI models. Please give me as complete code as possible, with instructions on what folders they should be placed in, and how to package them for installation into Discourse.

Assistant (gpt-40)

Tue May 20 2025 19:07:23 GMT-0400 (Eastern Daylight Time)

User wants to support multiple external AI model providers in their Discourse instance, including Google Gemini/Gemma, Groq-hosted models, Anthropic, and the full OpenAI model suite. They are requesting complete plugin code and folder structure for Discourse plugin development.

Tue May 20 2025 19:07:30 GMT-0400 (Eastern Daylight Time)

To support multiple AI providers (OpenAI, Anthropic, Google Gemini/Gemma, and Groq-hosted models) in Discourse using a custom plugin, you can build a plugin that extends the discourse-ai framework with modular provider support. Below is a complete working structure with setup instructions.

Overview

We will:

- Create a plugin called discourse-multi-ai
- Hook into Discourse AI to support:
- OpenAI (GPT-40, GPT-3.5-turbo, etc.)
- Anthropic (Claude 3 family)
- Google Gemini/Gemma
- Groq (Mixtral, LLaMA, etc.)

Plugin Structure

```
plugins/
 discourse-multi-ai/
    plugin.rb
>
     - lib/
└─ discourse<em>multi</em>ai/
         - engine.rb
providers/
>
           base<em>provider.rb
>
         - openai</em>provider.rb
>
             - gemini</em>provider.rb
>
           groq<em>provider.rb
>
     - config/
└─ locales/
>
         \sqsubseteq en.yml
```

Step-by-Step Setup

1. plugin.rb

```
ruby
<h1&gt;name: discourse-multi-ai&lt;/h1&gt;
<h1&gt;version: 0.1&lt;/h1&gt;
<h1&gt;authors: Tem Noon, ChatGPT&lt;/h1&gt;
<h1&gt;reguired</em>version: 3.2.0&lt;/h1&gt;
<h1&gt;about: Extend Discourse AI with multiple external AI
providers</h1&gt;
enabled<em>site</em>setting :multi<em>ai</em>enabled
require<em>relative "lib/discourse</em>multi<em>ai/engine"
after</em>initialize do
 require<em>relative "lib/discourse</em>multi<em>ai/providers/base</-</p>
em>provider"
 require<em>relative "lib/discourse</em>multi<em>ai/providers/openai</-</p>
em>provider"
 require<em>relative "lib/discourse</em>multi<em>ai/providers/-
anthropic</em>provider"
 require<em>relative "lib/discourse</em>multi<em>ai/providers/gemini</-</p>
em>provider"
 require<em>relative "lib/discourse</em>multi<em>ai/providers/groq</-</p>
em>provider"
 DiscourseAi::Completions::BaseCompletions.register<em>provider(:openai,
DiscourseMultiAi::OpenAiProvider)
 DiscourseAi::Completions::BaseCompletions.register</em>provider(:an-
thropic, DiscourseMultiAi::AnthropicProvider)
    DiscourseAi::Completions::BaseCompletions.register<em>provider(:gemini,
DiscourseMultiAi::GeminiProvider)
    DiscourseAi::Completions::BaseCompletions.register</em>provider(:groq,
DiscourseMultiAi::GrogProvider)
end
```

2. lib/discoursemultiai/engine.rb

```
ruby
module DiscourseMultiAi
 class Engine < ::Rails::Engine
 engine<em>name "discourse</em>multi<em>ai"
 isolate</em>namespace DiscourseMultiAi
 end
end
end
```

3. lib/discoursemultiai/providers/baseprovider.rb

```
ruby
module DiscourseMultiAi
 class BaseProvider
 def self.perform</em>completion(prompt:, model:, api<em>key:, **op-
tions)
 raise NotImplementedError
 end
 end
end
end
end
```

4. OpenAI Provider - openai provider.rb

```
ruby
require "net/http"
require "json"
module DiscourseMultiAi
 class OpenAiProvider < BaseProvider
 def self.perform<em>completion(prompt:, model:, api</em>key:, tempera-
ture: 0.7)
    uri = URI("https://api.openai.com/v1/chat/completions")
>
      headers = {}
>
        "Authorization" => "Bearer #{api<em>key}",
>
       "Content-Type" => "application/json"
>
      }
body = \{ 
>
       model: model,
>
       messages: [{ role: "user", content: prompt }],
>
       temperature: temperature
>
    }
>
     response = Net::HTTP.post(uri, body.to</em>json, headers)
>
      JSON.parse(response.body)["choices"].first["message"]["content"]
>
     end
 end
end
```

5. Anthropic Provider - anthropic provider.rb

```
ruby
require "net/http"
require "json"
module DiscourseMultiAi
 class AnthropicProvider < BaseProvider
 def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
    uri = URI("https://api.anthropic.com/v1/messages")
     headers = {}
>
      "x-api-key" => api</em>key,
       "anthropic-version" =&qt; "2023-06-01",
>
       "content-type" => "application/json"
>
 }
>
     body = \{ 
       model: model,
>
       max<em>tokens: 1024,
      messages: [{ role: "user", content: prompt }],
>
       temperature: temperature
>
>
    }
     response = Net::HTTP.post(uri, body.to</em>json, headers)
>
     JSON.parse(response.body).dig("content", 0, "text")
>
   end
 end
end
```

6. Gemini Provider - geminiprovider.rb

```
require "net/http"
require "json"
module DiscourseMultiAi
 class GeminiProvider < BaseProvider
     def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
     uri = URI("https://generativelanguage.googleapis.com/v1beta/models/#
{model}:generateContent?key=#{api</em>key}")
       body = \{ 
contents: [{ parts: [{ text: prompt }] }],
>
>
        generationConfig: { temperature: temperature }
      }
headers = { "Content-Type" => "application/json" }
<a>>
      response = Net::HTTP.post(uri, body.to<em>json, headers)
      JSON.parse(response.body).dig("candidates", 0, "content", "parts",
<a>>
0, "text") 
 end
 end
end
```

7. Groq Provider - groq provider.rb

```
require "net/http"
require "json"
module DiscourseMultiAi
 class GrogProvider < BaseProvider
def self.perform<em>completion(prompt:, model:, api</em>key:, tempera-
ture: 0.7)
      uri = URI("https://api.groq.com/openai/v1/chat/completions")
<q>
headers = {}
<q>>
        "Authorization" => "Bearer #{api<em>key}",
        "Content-Type" => "application/json"
>
      }
     body = \{ 
>
       model: model,
>
       messages: [{ role: "user", content: prompt }],
< < > >
        temperature: temperature
}
      response = Net::HTTP.post(uri, body.to</em>json, headers)
JSON.parse(response.body)["choices"].first["message"]["content"]
>
    end
 end
end
```

8. Admin Settings (Optional)

To wire this to the admin UI, create config/locales/en.yml:

```
yaml
 site<em>settings:
 multi</em>ai<em>enabled: "Enable Multiple AI Providers"
```

Add more settings via settings.yml if needed (e.g., per-provider API keys).

Testing & Installation

Local Plugin Installation

1. Place the folder in plugins/ of your Discourse installation:

```
 /var/discourse/plugins/discourse-multi-ai  <
```

1. Rebuild Discourse:

```
bash
 ./launcher rebuild app
<<p><<p><<p><</p>
```

- 1. Enable the plugin via Admin > Settings > Plugins
- 2. Test via custom usage: You'll want to wire these providers into specific AI tools (summarizers, classifiers, chatbots). That will require extending or subclassing tools from discourse-ai.

Bonus: Plug Providers into Custom AI Tools

You can use a pattern like this in a separate module:

```
ruby
module MyCurator
 def self.summarize(post)
 prompt = "Summarize: #{post.raw}"
 DiscourseMultiAi::AnthropicProvider.perform</em>completion(
 prompt: prompt,
 model: "claude-3-opus-20240229",
 api<em>key: SiteSetting.anthropic</em>api_key
 end
 end
end
```

✓Next Step

Would you like help:

- Wiring this into category-specific AI curators?
- Adding support for streaming outputs (where available)?
- Generating admin UI to control each provider?

Let me know which of these you'd like, and I'll expand this into a deployable plugin template.

Assistant (gpt-40)

Tue May 20 2025 19:07:35 GMT-0400 (Eastern Daylight Time)

User

Tue May 20 2025 19:13:05 GMT-0400 (Eastern Daylight Time)

Where is the plugins folder in the discourse installlation?

Tue May 20 2025 19:13:06 GMT-0400 (Eastern Daylight Time)

In a standard Discourse Docker installation (using the official launcher), the plugins folder is located inside the app container environment. Here's how you can access and use it:



Location of the Plugins Folder

If you're using the official Docker setup:

On your host machine (where Discourse is installed):

/p>/var/discourse/plugins/

Example:

bash

cd /var/discourse/plugins

This is where you can **clone or copy** custom plugins.

Installing Your Plugin

Method 1: Clone directly into plugins folder

bash

cd /var/discourse/plugins <qit clone https://qithub.com/yourusername/discourse-multi-ai.qit</p>

> Make sure your plugin has the correct plugin.rb structure and subfolders (as described earlier).

Rebuild the App

After placing your plugin in /var/discourse/plugins, you must rebuild the Discourse container to load it:

bash
cd /var/discourse
./launcher rebuild app

This step is essential—plugins are only loaded during the container build.

Werifying the Plugin

Once the container is rebuilt:

- Go to Admin > Plugins your plugin should be listed.
- You can also visit Admin > Settings > Plugins to configure it, if it exposes any settings.

X Alternative: Development (non-Docker) Setup

If you're running Discourse in a local development environment (non-Docker, for hacking):

bash
~/discourse/plugins/

Just drop your plugin there and restart your Puma server.

Let me know if you'd like a script to automate the cloning and registration of your plugin inside a Droplet setup.

Assistant (gpt-40)

Tue May 20 2025 19:13:15 GMT-0400 (Eastern Daylight Time)

User

Tue May 20 2025 19:16:15 GMT-0400 (Eastern Daylight Time)

I see. So while in my discourse folder there is no plugin folder, by creating one, and hooking it into my git account, I can install it into my docker install of discourse when I rebuild that docker container. Did I understand that right?

Tue May 20 2025 19:16:16 GMT-0400 (Eastern Daylight Time)

Yes — you've understood that almost perfectly, just a slight clarification needed. Here's the exact process, refined for clarity:

✓Your Understanding (Refined)

> "While in my discourse folder there is no plugin folder, by creating one, and hooking it into my git account, I can install it into my docker install of discourse when I rebuild that docker container."

✓ Corrected & Expanded Version:

• The actual Discourse app container lives inside a system-level directory, typically:

```
/var/discourse
```

- Inside that directory is where the **Docker launcher setup** lives including your container definition in containers/app.yml.
- The plugins folder is inside /var/discourse, and it's used during the container rebuild process to add any plugins to your Discourse instance.

₹ Step-by-Step Guide

1. Navigate to the plugins folder:

```
bash
cd /var/discourse/plugins
```

If it doesn't exist yet (rare, but can happen if you've never added plugins), just create it:

```
bash mkdir -p /var/discourse/plugins
```

2. Clone your plugin from GitHub:

```
bash
git clone https://github.com/YOUR<em>GITHUB</em>USER/discourse-multi-ai.-
git
```

Now your structure will look like:

```
\label{local-plugins} $$\p>/\p>$ is course-multi-ai/plugin.rb
```

3. Rebuild the container to load the plugin:

```
bash
cd /var/discourse
./launcher rebuild app
```

> This step is **critical** — plugins are only loaded when the container is rebuilt.

Extra Tip: Adding Git Repo Directly to app.yml

As an alternative to manually cloning into /plugins, you can also just add the plugin Git URL to your app.yml like this:

```
yaml
hooks:
 after<em>code:
 data-list-type="ul">exec:
    cd:
    shome/plugins
 <md:</p>
 data-list-type="ul">git clone https://github.com/YOUR</-em>GITHUB_USER/discourse-multi-ai.git
```

Then run ./launcher rebuild app .

This method is **especially useful** for provisioning or automated deployments.

Would you like me to generate a full plugin scaffold (plugin.rb, settings.yml, en.yml) that you can push to GitHub and pull into Discourse?

Tue May 20 2025 19:16:27 GMT-0400 (Eastern Daylight Time)

User

Tue May 20 2025 19:27:47 GMT-0400 (Eastern Daylight Time)

Yes, that sounds great. Note that my username on github is 'temnoon'.

```
Assistant (gpt-40)
                                        Tue May 20 2025 19:27:48 GMT-0400 (Eastern Daylight Time)
"name": "discourse-multi-ai",
"type": "code/ruby",
"content": "# name: discourse-multi-ai\n# about: Integrate multiple external AI
providers into Discourse AI (OpenAI, Anthropic, Gemini, Groq)\n# version: 0.1\n# au-
thors: Tem Noon, ChatGPT\n\# requiredversion: 3.2.0\n\# url: https://github.com/tem-
noon/discourse-multi-ai\n\nenabledsitesetting
                                                :multiaienabled\n\nrequirerelative
\"lib/discoursemultiai/engine\"\n\nafterinitialize do\n requirerelative \"lib/discoursemul-
tiai/providers/baseprovider\"\n requirerelative \"lib/discoursemultiai/providers/ope-
naiprovider\"\n requirerelative \"lib/discoursemultiai/providers/anthropicprovider\"\n
requirerelative \"lib/discoursemultiai/providers/geminiprovider\\\n requirerelative
\"lib/discoursemultiai/providers/groaprovider\"\n\n
DiscourseAi::Completions::BaseCompletions.registerprovider(:openai,
DiscourseMultiAi::OpenAiProvider)\n DiscourseAi::Completions::BaseCompletions.
registerprovider(:anthropic,
                                             DiscourseMultiAi::AnthropicProvider\\n
DiscourseAi::Completions::BaseCompletions.registerprovider(:gemini,
DiscourseMultiAi::GeminiProvider)\n
                                       DiscourseAi::Completions::BaseCompletions.
register_provider(:groq, DiscourseMultiAi::GroqProvider)\nend"
```

Tue May 20 2025 19:27:54 GMT-0400 (Eastern Daylight Time)

Great! I've scaffolded the core plugin.rb file for your discourse-multi-ai plugin. Next, here's how to complete the plugin with folder structure and required files for your GitHub repo at https://github.com/temnoon/discourse-multi-ai.

File Structure

```
discourse-multi-ai/
  plugin.rb
|-- lib/
| discourse<em>multi</em>ai/
      - engine.rb
|
      └─ providers/
|
        base<em>provider.rb
- openai</em>provider.rb
        - anthropic<em>provider.rb
|
         pemini</em>provider.rb
groq<em>provider.rb
|-- config/
| L locales/
|
     — en.yml
 settings.yml
```

Add These Files

lib/discourse multiai/engine.rb

```
ruby
module DiscourseMultiAi
 class Engine < ::Rails::Engine
 engine</em>name "discourse<em>multi</em>ai"
 isolate<em>namespace DiscourseMultiAi
 end
end
end
```

lib/discourse multiai/providers/baseprovider.rb

```
ruby
module DiscourseMultiAi
 class BaseProvider
 def self.perform<em>completion(prompt:, model:, api</em>key:, **options)
 raise NotImplementedError, "Subclasses must implement this"
 end
 end
end
end
end
```

Example Provider - openaiprovider.rb

You'll duplicate this template for each provider (gemini provider.rb, groqprovider.rb, etc.).

```
require "net/http"
require "json"
module DiscourseMultiAi
 class OpenAiProvider < BaseProvider
    def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7) 
     uri = URI("https://api.openai.com/v1/chat/completions")
>
>
      headers = {}
        "Authorization" => "Bearer #{api</em>key}",
>
        "Content-Type" => "application/json"
>
>
     }
     body = \{ 
>
       model: model,
       messages: [{ role: "user", content: prompt }],
<q>>
>
        temperature: temperature
>
      response = Net::HTTP.post(uri, body.to<em>json, headers)
>
      JSON.parse(response.body).dig("choices", 0, "message", "content")
>
     end
 end
end
```

Let me know if you'd like me to prefill the Anthropic, Gemini, and Groq versions here too.

config/locales/en.yml

```
yaml
en:
 site</em>settings:
 multi<em>ai</em>enabled: "Enable Multi AI Provider Plugin"
```

settings.yml

```
yaml
multi<em>ai
 default: false
 client: true
 type: bool
```

Deployment Instructions

Once you've added the above files:

1. Commit and push to GitHub:

```
bash
 git init
 git remote add origin https://github.com/temnoon/discourse-multi-
ai.git
 git add .
 git add .
 git commit -m "Initial commit of multi-provider AI plugin"
 git push -u origin main
```

1. Add this plugin to your Discourse app via:

1. Rebuild:

Would you like me to auto-generate the full provider files for Anthropic, Gemini, and Groq too?

User	Tue May 20 2025 20:21:24 GMT-0400 (Eastern Daylight Time)
OK, I have this directory, with the file contents you've given me. The yml files, however, are still empty.	
(base) tem@MacBookPro discourse % tree	
— discourse-multi-ai	
— config	
locales	
lib	
discourse-multi-ai	
engine.rb	
anthropicprovider.rb	
— geminiprovider.rb	
groqprovider.rb	
openai_provider.rb	
—— plugin.rb	
L—settings.yml	
7 directories, 9 files	

Tue May 20 2025 20:21:25 GMT-0400 (Eastern Daylight Time)

Perfect — your folder structure is exactly right. Now, let me give you the full contents for the two YAML files you're missing: en.yml and settings.yml.



config/locales/en.yml

```
yaml
< discourse<em>multi</em>ai:
< plugin<em>name: "Multi AI Provider Plugin"
 plugin</em>description: "Adds support for OpenAI, Anthropic, Gemini, and Groq to Discourse AI"
 site<em>settings:
 multi</em>ai<em>enabled: "Enable Multi AI Provider Plugin"
 openai</em>api<em>key: "OpenAI API Key"
 anthropic</em>api<em>key: "Anthropic API Key"
 gemini</em>api<em>key: "Google Gemini API Key"
 groq</em>api<em>key: "Groq API Key"
 groq</em>api<em>key: "Groq API Key"
```

m u settings.yml

```
yaml
multi</em>ai<em>enabled:
 default: false
  client: true
 type: bool
openai</em>api<em>key:
 default: ""
 secret: true
 client: false
 type: string
anthropic</em>api<em>key:
 default: ""
 secret: true
  client: false
 type: string
gemini</em>api<em>key:
 default: ""
 secret: true
 client: false
 type: string
groq</em>api key:
 default: ""
 secret: true
  client: false
 type: string
```

These settings will expose your API key inputs in Admin > Settings > Plugins, scoped to your plugin.

Final Steps

1. Add the YAML files to Git:

```
bash
 git add config/locales/en.yml settings.yml
 git commit -m "Add plugin settings and localization"
```

1. Push to GitHub:

```
bash
 git push
<<p><<p><</p>
```

1. Rebuild Discourse on your droplet:

```
bash
 cd /var/discourse
 ./launcher rebuild app
<<p><<p><<p><<p><<p>
```

Would you now like me to help wire these provider classes into actual AI summarization or conversation features (e.g., per-category AI Curator agents)?

User

Tue May 20 2025 20:24:01 GMT-0400 (Eastern Daylight Time)

I added the yml's, but I haven't set up git for it yet. Give me that command again, and we'll push the whole thing.

Tue May 20 2025 20:24:02 GMT-0400 (Eastern Daylight Time)

Excellent — here's a quick set of commands to initialize a Git repository, commit your plugin code, and push it to GitHub under your tempoon account.

X Git Initialization and Push Instructions

Run these commands from inside your discourse-multi-ai folder:

bash
cd /path/to/discourse-multi-ai

(Replace with the actual path; e.g., ~/discourse/discourse-multi-ai)

1. Initialize the Git repository

bash
git init

2. Add your remote GitHub repository

If you haven't created the GitHub repo yet, do that now at:

https://github.com/new

Name it discourse-multi-ai

Then connect it:

bash

<git remote add origin https://github.com/temnoon/discourse-multi-ai.git</p>

3. Add all plugin files

```
bash git add .
```

4. Commit the initial version

```
bash git commit -m "Initial commit of discourse-multi-ai plugin with multi-provider support"
```

5. Push to GitHub

```
bash
git branch -M main
git push -u origin main
```

Once that's done, you'll be able to add the plugin to your Discourse server like this in app.yml:

```
yaml
hooks:
 after_code:
 data-list-type="ul">exec:
    cd:
$home/plugins
 cmd:
 data-list-type="ul">git clone https://github.com/-temnoon/discourse-multi-ai.git
```

Then run:

```
bash
cd /var/discourse
./launcher rebuild app
```

Let me know if you want a .gitignore, README.md, or GitHub Actions workflow to help automate releases or linting.

User

Tue May 20 2025 20:32:20 GMT-0400 (Eastern Daylight Time)

The push needs to use ssh, not password ...

Tue May 20 2025 20:32:21 GMT-0400 (Eastern Daylight Time)

Perfect - to push using SSH instead of HTTPS, you'll just need to:

- 1. Make sure your SSH key is added to your GitHub account
- 2. Use the SSH version of your remote URL

Step-by-Step: Push to GitHub via SSH

1. Confirm SSH key exists on your system

Check for an existing key:

```
bash
ls ~/.ssh/id<em>rsa.pub
```

If it doesn't exist, generate one:

```
bash ssh-keygen -t rsa -b 4096 -C "your-email@example.com"
```

Then copy the public key:

```
bash
cat ~/.ssh/id</em>rsa.pub
```

Go to your GitHub SSH settings \rightarrow New SSH key \rightarrow paste the contents.

2. Update Git Remote to use SSH

If you've already added the HTTPS remote, change it:

```
bash
git remote set-url origin git@github.com:temnoon/discourse-multi-ai.git
```

If you haven't added a remote yet:

hash

git remote add origin git@github.com:temnoon/discourse-multi-ai.git

3. Push via SSH

Now push like this:

bash
git push -u origin main

> You might be prompted to allow the SSH connection or enter your SSH key passphrase (if you set one).

Would you like me to generate a clean README.md now for your GitHub project as well?

User

Tue May 20 2025 20:33:49 GMT-0400 (Eastern Daylight Time)

Yes, good idea.

Tue May 20 2025 20:33:51 GMT-0400 (Eastern Daylight Time)

Here's a clean and informative README.md you can use for your discourse-multi-ai GitHub repo. It's structured to be both user-friendly and developer-readable.



README . md markdown <h1>discourse-multi-ai</h1> Multi-provider AI integration plugin for Discourse This plugin adds support for multiple external AI providers into the <a</p> href="https://github.com/discourse/discourse-ai">Discourse AI plugin, including: data-list-type="ul">♥ OpenAI (GPT-40, GPT-3.5, DALL-E, etc.) data-list-type="ul">♥ Anthropic (Claude 3 models)<li data-listtype="ul"> Google Gemini / Gemmadata-list-type="ul"> Grog-hosted open source models (Mixtral, LLaMA, etc.) @temnoon <hr/> <h2> Features</h2> data-list-type="ul">Extend Discourse AI with modular, pluggable provider supportdata-list-type="ul">Use your own API keysli data-list-type="ul">Select model and provider dynamicallydata-listtype="ul">Designed to integrate with AI-powered tools like summarization, chat, and moderationdata-list-type="ul">Easy to expand - add new providers by subclassing <code>BaseProvider</code><hr/> <h2> Installation</h2> Add this plugin to your Discourse installation: SSH into your server and edit the <code>containers/app.yml</code> filedata-list-type="ol">Add the following under

<code>hooks > aftercode</code>:

yaml

hooks:

aftercode:

exec:

```
cd: $home/plugins
```

cmd:

• git clone git@github.com:temnoon/discourse-multi-ai.git

```
Rebuild your Discourse instance:
```

bash

cd /var/discourse

./launcher rebuild app

```
<hr/&gt;
&lt;h2&gt;  Configuration&lt;/h2&gt;
In the Discourse Admin panel, go to:
xtrong>Admin &gt; Settings &gt; Plugins &gt; Multi AI Provider Plugin
Fill in the API keys for any of the supported providers:
full data-list-type="ul"><code>openai<em>api</em>key</code>data-list-type="ul"><code>anthropic<em>api</em>key</code>data-list-type="ul"><code>gemini<em>api</em>key</code>data-list-type="ul"><code>gengini<em>api</em>key</code>data-list-type="ul"><code>groq<em>api</em>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code>data-list-type="ul"><code>groq<em>api</m>key</code></m>data-list-type="ul"><code>groq<em>api</m>key</code></m>data-list-type="ul"><code>groq<em>api</m></m>em>key</m>code>groq<em>api</m>em>key</m>code>groq<em>api</m>em>key</m>code>groq<em>api</m>em>key</m>code>groq<em>api</m>em>key</m>code>groq<em>api</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key</m>em>key
```

ruby

DiscourseMultiAi::OpenAiProvider.performcompletion(

```
prompt: "Summarize this...",
```

model: "gpt-40",

apikey: SiteSetting.openaiapikey

```
Each provider handles its own API calls and responses. See the <code>/-
  lib/discourse<em>multi</em>ai/providers</code> folder for examples.
  <hr/&gt;
  <h2&gt; X Extending&lt;/h2&gt;
  Add a new provider by subclassing <code>BaseProvider</code> and implement-
  ing <code>.perform<em>completion</code>.
ruby
class\ My Provider < Base Provider
def self.performcompletion(prompt:, model:, api_key:, **opts)
# Your API call here
end
end
  Then register it inside <code>plugin.rb</code>.
  <hr/&gt;
  <h2&gt; License&lt;/h2&gt;
  MIT
  <hr/&gt;
  <h2&gt; Acknowledgments&lt;/h2&gt;
  data-list-type="ul">Built to extend the amazing <a href="https://-</a>
  github.com/discourse/discourse-ai">discourse-ai</a> framework<li data-
  list-type="ul">Inspired by modular plugin architectures in open-source LLM
  hosting platforms
```

Let me know if you'd like a GitHub Actions CI workflow, versioning badge, or automated plugin tests scaffold as well.

User

Tue May 20 2025 21:39:43 GMT-0400 (Eastern Daylight Time)

When I rebuilt, it aborted with these errors, related to our new plugin.

FAILED

Pups::ExecError: cd /var/www/discourse/plugins && git clone git@github.com:tem-noon/discourse-multi-ai.git failed with return #<Process::Status: pid 258 exit 128>

Location of failure: /usr/local/lib/ruby/gems/3.3.0/gems/pups-1.2.1/lib/pups/execcommand.rb:132:in `spawn'

exec failed with the params {"cd"=>"\$home/plugins", "cmd"=>["git clone https://github.com/discourse/dockermanager.git", "git clone https://github.com/discourse-ai.git", "git clone git@github.com:temnoon/discourse-multi-ai.git"]}

bootstrap failed with exit code 128

FAILED TO BOOTSTRAP please scroll up and look for earlier error messages, there may be more than one.

/discourse-doctor may help diagnose the problem.

053032fa0dc2fdcef4e7121fad9a2d18bc3e17aaa8189bd08633f3ecdd3e6a5b

root@social-humanizer-mastodon:/var/discourse#

Tue May 20 2025 21:39:44 GMT-0400 (Eastern Daylight Time)

Thanks for the detailed error log - the issue is clear:

> X git clone git@github.com:temnoon/discourse-multi-ai.git failed with return code 128

This error happens because the **Discourse rebuild process** (inside **Docker**) does not have access to your personal SSH key, which is required for cloning via git@github.com: URLs.

Solution: Use HTTPS for Plugin Clones in

app.yml

Inside the Docker container, SSH-based Git access won't work unless you go through a lot of extra SSH key setup, so the easiest and most reliable fix is to switch to the public HTTPS version of the repo in your app.yml:

*Fix your containers/app.yml

Change this:

yaml
qoit clone git@github.com:temnoon/discourse-multi-ai.git

To this:

yaml
qoit clone https://github.com/temnoon/discourse-multi-ai.git

Then Rebuild:

bash
cd /var/discourse
./launcher rebuild app

This will now succeed because HTTPS Git cloning does not require credentials for public repositories.

Optional: Want to use SSH anyway?

If you really want SSH access for some reason (e.g., private plugin repo):

- 1. You would need to mount your SSH key into the container (discouraged for security).
- 2. Or create a **deploy key** on GitHub and inject it via Docker volume mounts or ENV vars.
- > But again: not worth it for public plugins. Use HTTPS.

Let me know once that rebuild works — we can then move on to testing whether all providers initialize properly or not.

User

Tue May 20 2025 21:53:05 GMT-0400 (Eastern Daylight Time)

I changed the ssh to http, but it still failed. I ran a further test with \(\textit{J}\)discourse-doctor which also failed. Tell me if there is anything specific you can use from that log \(\text{...}\) it's very long.

FAILED

Pups::ExecError: cd /var/www/discourse && su discourse -c 'bundle exec rake db:mi-grate' failed with return #<Process::Status: pid 874 exit 1>

Location of failure: /usr/local/lib/ruby/gems/3.3.0/gems/pups-1.2.1/lib/pups/execcom-mand.rb:132:in `spawn'

exec failed with the params {"cd"=>"\$home", "tag"=>"migrate", "hook"=>"dbmigrate", "cmd"=>["su discourse -c 'bundle exec rake db:migrate'"]}

bootstrap failed with exit code 1

FAILED TO BOOTSTRAP please scroll up and look for earlier error messages, there may be more than one.

/discourse-doctor may help diagnose the problem.

90b9c5a756c6a3ba960c7de29fa43136d9fe8bc3554919144086caee21551738

Tue May 20 2025 21:53:07 GMT-0400 (Eastern Daylight Time)

Thanks — you're past the **Git clone issue**, so now the problem is deeper in the **Rails** boot process, specifically during the rake db:migrate step. The actual cause is usually not visible in the final lines but earlier in the logs.

What We Know So Far

- • The plugin cloned successfully via HTTPS.
- X During db:migrate, it failed with:
- > bundle exec rake db:migrate failed with exit 1

That strongly suggests one of the following:

- 1. Your plugin references a constant or gem that is not loaded yet.
- 2. A syntax error or missing dependency exists in one of your plugin files.
- 3. A fatal error occurred when loading plugin.rb, engine.rb, or a provider.

X Next Steps: Find the First Ruby Exception

The output you posted:

Location of failure: /usr/local/lib/ruby/gems/3.3.0/gems/pups-1.2.1/lib/-pups/execcommand.rb:132:in <code>spawn'

...is a wrapper error. To diagnose properly, scroll up in your rebuild output and look for something like:

- NameError
- LoadError
- NoMethodError
- undefined method
- uninitialized constant
- or even a SyntaxError

These will usually be in red and mention the plugin file that triggered it.

Quick Fix: Run Manual Plugin Syntax Check

If you're still on the server:

```
bash
cd /var/discourse/plugins/discourse-multi-ai
ruby -c plugin.rb
```

Do the same for other Ruby files:

```
bash
find lib -name '*.rb' -exec ruby -c {} \;
```

That will validate Ruby syntax and catch any obvious mistake (like invalid YAML interpolation or a typo in a require path).

Common Issues to Check in Our Plugin

Here are a few known error traps from earlier that may have slipped in:

1. Incorrect folder name reference in requirerelative

Make sure this line in plugin.rb:

```
require<em>relative "lib/discourse</em>multi<em>ai/engine"
...matches the actual folder name:
 lib/discourse-multi-ai
If the folder is named with dashes ( - ) rather than underscores
( ), Ruby will throw LoadError .
Fix: Either rename the folder to lib/discoursemultiai/ (recommend-
ed), or update the requirerelative paths to use the dash:
 require</em>relative "lib/discourse-multi-ai/engine"
2. Gem dependencies
If your plugin calls APIs using Net::HTTP or json , that's okay
```

If your plugin calls APIs using Net::HTTP or json, that's okay — those are standard libs. But if you use anything not included by default in Discourse (like faraday, httparty, etc.), you'd need to declare those gems, and they must be installed during bootstrap (which is not ideal unless critical).

Temporary Disable Your Plugin (for Debugging) If you're blocked and need your forum back up fast, comment out the plugin in your app.yml:

Then:

bash ./launcher rebuild app

Once it's back online, you can iterate locally and test the plugin in isolation via $bundle\ exec\ rails\ c$ inside a devenvironment.

Would you like to paste the full top of the error log (from just before rake db:migrate` fails)? I can pinpoint the offending line quickly. Or, if you'd like, I can generate a version of the plugin that is fully validated to pass boot.

User

Tue May 20 2025 21:57:53 GMT-0400 (Eastern Daylight Time)

Here is a load error:

871:C 21 May 2025 01:49:37.473 # oO0OoO0OoO Redis is starting oO0OoO0OoO

871:C 21 May 2025 01:49:37.473 # Redis version=7.0.15, bits=64, commit=00000000, modified=0, pid=871, just started

871:C 21 May 2025 01:49:37.474 # Configuration loaded

871:M 21 May 2025 01:49:37.475 monotonic clock: POSIX clockgettime

871:M 21 May 2025 01:49:37.475 # Warning: Could not create server TCP listening socket :6379: bind: Address already in use

871:M 21 May 2025 01:49:37.476 # Failed listening on port 6379 (TCP), aborting.

I, [2025-05-21T01:49:47.461026 #1] INFO --:

I, [2025-05-21T01:49:47.461978 #1] INFO -- : > cd /var/www/discourse && su discourse -c 'bundle exec rake db:migrate'

rake aborted!

LoadError: cannot load such file -- /var/www/discourse/plugins/discourse-multi-ai/lib/discoursemultiai/engine (LoadError)

/var/www/discourse/plugins/discourse-multi-ai/plugin.rb:10:in require relative'

/var/www/discourse/plugins/discourse-multi-ai/plugin.rb:10:in activate!

/var/www/discourse/lib/plugin/instance.rb:806:in instanceeval'

/var/www/discourse/lib/plugin/instance.rb:806:in activate!'

```
/var/www/discourse/lib/discourse.rb:347:in block in activate plugins!'
/var/www/discourse/lib/discourse.rb:344:in each'
/var/www/discourse/lib/discourse.rb:344:in activateplugins!'
/var/www/discourse/config/application.rb:193:in
                                                             block
                                                                        in
<class:Application>'
/var/www/discourse/lib/plugin.rb:6:in initialization guard'
/var/www/discourse/config/application.rb:193:in <class:Application>'
/var/www/discourse/config/application.rb:62:in <module:Discourse>'
/var/www/discourse/config/application.rb:61:in <top (required)>'
/var/www/discourse/Rakefile:7:in <top (required)>'
/var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/exe/rake:27:in <top (required)>'
/usr/local/bin/bundle:25:in load'
 /usr/local/bin/bundle:25:in <main>'
(See full trace by running task with --trace)
```

Tue May 20 2025 21:57:54 GMT-0400 (Eastern Daylight Time)

Perfect - thank you! The problem is exactly what we suspected:

> LoadError: cannot load such file --

/var/www/discourse/plugins/discourse-multi-ai/lib/discoursemultiai/engine

This is caused by a mismatch in folder naming between:

- the folder on disk (lib/discourse-multi-ai/)
- and the code trying to load lib/discoursemultiai/engine.rb





Option 1 (Recommended): Rename the Folder

Rename your plugin's lib/discourse-multi-ai/ folder to use underscores instead of hyphens:

```
bash
cd /path/to/discourse-multi-ai/lib
mv discourse-multi-ai discourse<em>multi</em>ai
```

After that, your file layout will look like:

This will now match what plugin.rb expects:

```
ruby
require</em>relative "lib/discourse<em>multi</em>ai/engine"
```

Then try again:

bash
cd /var/discourse
./launcher rebuild app

Noption 2: Change requirerelative paths (not recommended)

If for some reason you want to keep the folder name with hyphens (which is unusual in Ruby), you'd have to rewrite all require relative paths to use "discourse-multi-ai" instead of "discourse-multiai", e.g.:

```
ruby
require_relative "lib/discourse-multi-ai/engine"
```

But this goes against Ruby conventions, so **Option 1** (renaming the folder) is the proper fix.

Let me know when you've renamed and rebuilt — then we can verify the plugin loads cleanly and begin testing provider calls!

User

Tue May 20 2025 21:59:59 GMT-0400 (Eastern Daylight Time)

Help me with the git commit and push commands

Tue May 20 2025 22:00:00 GMT-0400 (Eastern Daylight Time)

Absolutely — once you've made the fix (renamed lib/discourse-multi-ai → lib/discoursemultiai), here's how to commit and push that change to GitHub.

Quick Recap of What You Did

You renamed a folder, which Git needs to register as a change in your repo.

Git Commands to Commit and Push

> git add -A ensures Git tracks renames and deletions properly.

Once that's pushed, re-run:

```
bash
cd /var/discourse
./launcher rebuild app
```

Let me know when you're ready to move on to testing completions or adding actual category-level AI Curators!

User

Tue May 20 2025 22:08:03 GMT-0400 (Eastern Daylight Time)

A different error I think:

FAILED

Pups::ExecError: cd /var/www/discourse && su discourse -c 'bundle exec rake db:mi-grate' failed with return #<Process::Status: pid 873 exit 1>

Location of failure: /usr/local/lib/ruby/gems/3.3.0/gems/pups-1.2.1/lib/pups/execcommand.rb:132:in `spawn'

exec failed with the params {"cd"=>"\$home", "tag"=>"migrate", "hook"=>"dbmigrate", "cmd"=>["su discourse -c 'bundle exec rake db:migrate"]}

bootstrap failed with exit code 1

FAILED TO BOOTSTRAP please scroll up and look for earlier error messages, there may be more than one.

/discourse-doctor may help diagnose the problem.

2e3295d72aec231c0f50cae3c086cffb600ab276764aa8e767c6964e52c54fb0

Tue May 20 2025 22:08:04 GMT-0400 (Eastern Daylight Time)

Thanks — you're right: that's the same general stage (rake db:migrate), but a different failure is now likely causing it, since we've resolved the earlier requirerelative LoadError.

To proceed, we need the actual error message just above this line:

> Pups::ExecError: cd /var/www/discourse && su discourse -c 'bun-dle exec rake db:migrate'

That part is just the wrapper. You'll want to scroll up in your terminal or copy-paste the output from about **20–50 lines above that** point, especially anything that:

- Mentions discourse-multi-ai
- Contains words like NameError, undefined method, NoMethodError, or SyntaxError
- Comes from inside a plugin.rb, provider class, or initializer

What to Look For

Specifically, scan for lines that look like this:

```
text
NameError: uninitialized constant DiscourseMultiAi::GeminiProvider
```

or

```
text
SyntaxError: unexpected keyword</em>end, expecting end-of-input
```

or

```
text
undefined method `register<em>provider' for ...
```

If You Want to Manually Test

While waiting, here's a shortcut you can use to **test just the plugin logic** without needing to rebuild the container every time:

- 1. Run a local development environment (on your Mac)
- 2. Load the plugin in a checkout of Discourse
- 3. From the Discourse root:

```
bash
 bundle exec rails c
<</p>
```

Then inside the console:

```
ruby
cp>DiscourseMultiAi::OpenAiProvider.perform</em>completion(
 prompt: "Say hello",
 model: "gpt-3.5-turbo",
 api_key: "your-test-api-key"
(p>)
```

This will let you catch logic bugs without waiting for full rebuilds.

Would you like me to generate a fully minimal plugin version that just loads the providers (without trying to register into Discourse AI) — as a sanity check for isolated boot testing? Or just paste the upper log section so I can fix directly.

User Tue May 20 2025 22:11:33 GMT-0400 (Eastern Daylight Time)

The answer is in here somewhere ...

I, [2025-05-21T02:06:44.282732 #1] INFO -- : > cd /var/www/discourse && su discourse -c 'LOADPLUGINS=0 bundle exec rake plugin:pullcompatibleall'

I, [2025-05-21T02:06:50.007184 #1] INFO -- : discourse-ai is already at latest compatible version

discourse-multi-ai is already at latest compatible version

dockermanager is already at latest compatible version

I, [2025-05-21T02:06:50.008367 #1] INFO -- : > exec chpst -u redis -U redis /usr/bin/redis-server /etc/redis/redis.conf

I, [2025-05-21T02:06:50.011894 #1] INFO -- : > sleep 10

870:C 21 May 2025 02:06:50.034 # oO0OoO0OoOo Redis is starting oO0OoO0OoOo

870:C 21 May 2025 02:06:50.035 # Redis version=7.0.15, bits=64, commit=00000000, modified=0, pid=870, just started

870:C 21 May 2025 02:06:50.035 # Configuration loaded

870:M 21 May 2025 02:06:50.036 monotonic clock: POSIX clockgettime

870:M 21 May 2025 02:06:50.037 # Warning: Could not create server TCP listening socket :6379: bind: Address already in use

870:M 21 May 2025 02:06:50.038 # Failed listening on port 6379 (TCP), aborting.

I, [2025-05-21T02:07:00.021081 #1] INFO -- :

I, [2025-05-21T02:07:00.022467 #1] INFO -- : > cd /var/www/discourse && su discourse -c 'bundle exec rake db:migrate'

81: from /usr/local/bin/bundle:25:in <main>'

```
80: from /usr/local/bin/bundle:25:in load'
79: from /usr/local/lib/ruby/gems/3.3.0/gems/bundler-2.6.4/exe/bundle:20:in <top
(required) > '
78:
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
            from
2.6.4/lib/bundler/friendlyerrors.rb:117:in withfriendlyerrors'
77: from /usr/local/lib/ruby/gems/3.3.0/gems/bundler-2.6.4/exe/bundle:28:in block
in <top (required)>'
76:
            from
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
2.6.4/lib/bundler/cli.rb:29:in start'
75:
                                     /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
                 from
2.6.4/lib/bundler/vendor/thor/lib/thor/base.rb:584:in start'
 74:
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
            from
2.6.4/lib/bundler/cli.rb:35:in dispatch'
73:
                 from
                                     /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
2.6.4/lib/bundler/vendor/thor/lib/thor.rb:538:in dispatch'
            from
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
 72:
2.6.4/lib/bundler/vendor/thor/lib/thor/invocation.rb:127:in
                                                                         in-
vokecommand'
                                     /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
71:
                 from
2.6.4/lib/bundler/vendor/thor/lib/thor/command.rb:28:in run'
```

```
70:
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
2.6.4/lib/bundler/cli.rb:452:in exec'
69: from /usr/local/lib/ruby/gems/3.3.0/gems/bundler-2.6.4/lib/bundler/cli/exec.rb:23:in
run'
68:
                        /usr/local/lib/ruby/gems/3.3.0/gems/bundler-
2.6.4/lib/bundler/cli/exec.rb:59:in kernelload'
67: from /usr/local/lib/ruby/gems/3.3.0/gems/bundler-2.6.4/lib/bundler/cli/exec.rb:59:in
load'
 66:
              from
                             /var/www/discourse/vendor/bundle/ruby/3.
3.0/bin/rake:25:in <top (required)>'
65: from /var/www/discourse/vendor/bundle/ruby/3.3.0/bin/rake:25:in load!
 64: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/exe/rake:27:in <top (required)>'
63: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/appli-
cation.rb:80:in run'
 62: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/application.rb:214:in standardexceptionhandling'
61: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/appli-
cation.rb:83:in block in run'
 60: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/application.rb:132:in toplevel'
```

```
59: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/application.rb:147:in run withthreads'
```

- 58: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/application.rb:138:in block in toplevel'
- 57: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/application.rb:138:in each'
- 56: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/application.rb:138:in block(2 levels) in toplevel'
- 55: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/application.rb:188:in invoke task'
- 54: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.rb:188:in invoke'
- 53: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task. rb:199:in invokewithcallchain'
- 52: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.rb:199:in synchronize'
- 51: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task. rb:218:in block in invoke withcallchain'
- 50: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.rb:241:in invokeprerequisites'
- 49: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task. rb:241:in each'

```
48: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:243:in block in invokeprerequisites'
47: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:199:in invokewithcallchain'
46: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:199:in synchronize'
45: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:218:in block in invoke withcallchain'
44: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:241:in invokeprerequisites'
43: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:241:in each'
42: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:243:in block in invokeprerequisites'
41: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:199:in invokewithcallchain'
40: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:199:in synchronize'
39: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:219:in block in invoke withcallchain'
```

```
38: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:281:in execute'
37: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-13.2.1/lib/rake/task.
rb:281:in each'
36: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rake-
13.2.1/lib/rake/task.rb:281:in block in execute'
35:
                      /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/railties-
          from
7.2.2.1/lib/rails/application.rb:559:in block in runtasksblocks'
34: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rail-
ties-7.2.2.1/lib/rails/application.rb:411:in requireenvironment!'
33:
                     /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/zeitwerk-
          from
2.7.2/lib/zeitwerk/coreext/kernel.rb:34:in require'
32: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/boot-
snap-1.18.6/lib/bootsnap/loadpathcache/coreext/kernelrequire.
rb:30:in require'
31: from /usr/local/lib/ruby/3.3.0/bundledgems.rb:69:in block (2 levels) in
replace require'
30: from /usr/local/lib/ruby/3.3.0/bundledgems.rb:69:in require'
29: from /var/www/discourse/config/environment.rb:7:in <main>'
28: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rail-
ties-7.2.2.1/lib/rails/application.rb:435:in initialize!'
```

```
27:
          from
                       /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/railties-
7.2.2.1/lib/rails/initializable.rb:60:in run initializers'
26: from /usr/local/lib/ruby/3.3.0/tsort.rb:208:in tsorteach'
25: from /usr/local/lib/ruby/3.3.0/tsort.rb:229:in tsort each'
24: from /usr/local/lib/ruby/3.3.0/tsort.rb:350:in eachstronglycon-
nectedcomponent'
23: from /usr/local/lib/ruby/3.3.0/tsort.rb:350:in call'
22: from /usr/local/lib/ruby/3.3.0/tsort.rb:350:in each'
21: from /usr/local/lib/ruby/3.3.0/tsort.rb:352:in block in each stronglycon-
nectedcomponent'
20: from /usr/local/lib/ruby/3.3.0/tsort.rb:434:in eachstronglycon-
nectedcomponentfrom'
19: from /usr/local/lib/ruby/3.3.0/tsort.rb:353:in block (2 levels)
eachstronglyconnectedcomponent'
             /usr/local/lib/ruby/3.3.0/tsort.rb:231:in
18:
                                                                 block in
tsorteach'
17:
                       /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/railties-
7.2.2.1/lib/rails/initializable.rb:61:in block in runinitializers'
16: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rail-
ties-7.2.2.1/lib/rails/initializable.rb:32:in run'
```

- 15: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/railties-7.2.2.1/lib/rails/initializable.rb:32:in instance exect
- 14: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/rail-ties-7.2.2.1/lib/rails/application/finisher.rb:94:in block in <module:Finisher>'
- 13: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/activesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:77:in run loadhooks'
- 12: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/ac-tivesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:77:in each'
- 11: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/activesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:78:in block in run loadhooks'
- 10: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/ac-tivesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:92:in executehook'
- 9: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/activesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:87:in withexecutioncontrol'
- 8: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/ac-tivesupport-7.2.2.1/lib/activesupport/lazyloadhooks.rb:94:in block in executehook'
- 7: from /var/www/discourse/config/application.rb:208:in block in <class:Application>'
- 6: from /var/www/discourse/lib/plugin.rb:6:in initializationguard'

```
5: from /var/www/discourse/config/application.rb:208:in block (2 levels) in <class:Application>'
```

- 4: from /var/www/discourse/config/application.rb:208:in each'
- 3: from /var/www/discourse/lib/plugin/instance.rb:634:in notify afterinitial-ize'
- 2: from /var/www/discourse/lib/plugin/instance.rb:634:in each'
- 1: from /var/www/discourse/lib/plugin/instance.rb:636:in block in notify afterinitialize'

/var/www/discourse/plugins/discourse-multi-ai/plugin.rb:19:in block in activate!': uninitialized constant DiscourseAi::Completions::BaseCompletions (NameError)

INCOMPATIBLE PLUGIN

You are unable to start Discourse due to errors in the plugin at

/var/www/discourse/plugins/discourse-multi-ai

Please try removing this plugin and rebuilding again!

I, [2025-05-21T02:07:16.326288 #1] INFO -- : gem install tokenizers -v 0.5.4 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed tokenizers-0.5.4-x8664-linux

1 gem installed

gem install tiktokenruby -v 0.0.11.1 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed tiktokenruby-0.0.11.1-x8664-linux

1 gem installed

gem install ed25519 -v 1.2.4 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 -- no-document --ignore-dependencies --no-user-install

Building native extensions. This could take a while...

Successfully installed ed25519-1.2.4

1 gem installed

gem install Ascii85 -v 2.0.1 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed Ascii85-2.0.1

1 gem installed

gem install ruby-rc4 -v 0.1.5 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 -- no-document --ignore-dependencies --no-user-install

Successfully installed ruby-rc4-0.1.5

1 gem installed

gem install hashery -v 2.1.2 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 -- no-document --ignore-dependencies --no-user-install

Successfully installed hashery-2.1.2

1 gem installed

gem install ttfunk -v 1.8.0 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed ttfunk-1.8.0

1 gem installed

gem install afm -v 0.2.2 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed afm-0.2.2

1 gem installed

gem install pdf-reader -v 2.14.1 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed pdf-reader-2.14.1

1 gem installed

I, [2025-05-21T02:07:16.331357 #1] INFO --: Terminating async processes

I, [2025-05-21T02:07:16.331885 #1] INFO -- : Sending INT to HOME=/var/lib/postgresql USER=postgres exec chpst -u postgres:postgres:ssl-cert -U postgres:postgres:ssl-cert /usr/lib/postgresql/15/bin/postmaster -D /etc/postgresql/15/main pid: 41

2025-05-21 02:07:16.332 UTC [41] LOG: received fast shutdown request

I, [2025-05-21T02:07:16.332603 #1] INFO -- : Sending TERM to exec chpst -u redis -U redis /usr/bin/redis-server /etc/redis/redis.conf pid: 108

108:signal-handler (1747793236) Received SIGTERM scheduling shutdown...

2025-05-21 02:07:16.336 UTC [41] LOG: aborting any active transactions

2025-05-21 02:07:16.347 UTC [41] LOG: background worker "logical replication launcher" (PID 55) exited with exit code 1

2025-05-21 02:07:16.349 UTC [50] LOG: shutting down

2025-05-21 02:07:16.351 UTC [50] LOG: checkpoint starting: shutdown immediate

2025-05-21 02:07:16.364 UTC [50] LOG: checkpoint complete: wrote 5 buffers (0.0%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.009 s, sync=0.001 s, total=0.015 s; sync files=4, longest=0.001 s, average=0.001 s; distance=8 kB, estimate=8 kB

2025-05-21 02:07:16.373 UTC [41] LOG: database system is shut down

108:M 21 May 2025 02:07:16.381 # User requested shutdown...

108:M 21 May 2025 02:07:16.381 Saving the final RDB snapshot before exiting.

108:M 21 May 2025 02:07:16.428 DB saved on disk

108:M 21 May 2025 02:07:16.428 # Redis is now ready to exit, bye bye...

FAILED

Pups::ExecError: cd /var/www/discourse && su discourse -c 'bundle exec rake db:mi-grate' failed with return #<Process::Status: pid 873 exit 1>

Location of failure: /usr/local/lib/ruby/gems/3.3.0/gems/pups-1.2.1/lib/pups/execcommand.rb:132:in `spawn'

exec failed with the params {"cd"=>"\$home", "tag"=>"migrate", "hook"=>"db_migrate", "cmd"=>["su discourse -c 'bundle exec rake db:migrate'"]}

bootstrap failed with exit code 1

FAILED TO BOOTSTRAP please scroll up and look for earlier error messages, there may be more than one.

/discourse-doctor may help diagnose the problem.

2e3295d72aec231c0f50cae3c086cffb600ab276764aa8e767c6964e52c54fb0

Tue May 20 2025 22:11:34 GMT-0400 (Eastern Daylight Time)



You've now identified the real cause of the current failure:

uninitialized constant

DiscourseAi::Completions::BaseCompletions (NameError)

> from: plugin.rb:19:in 'block in activate!'



You're trying to use a class:

```
ruby
DiscourseAi::Completions::BaseCompletions
```

But this class doesn't exist (or isn't yet loaded) in the plugin you're writing.

Why?

- discourse-ai does not export BaseCompletions as a stable public class.
- You may have mistaken it for a pluggable registry system, but it's actually not structured that way yet.

X Solution: Safely Integrate Without Assuming

Internals

We should restructure your plugin to avoid referencing BaseCompletions at all unless it's clearly exposed.

Instead, you can:

- Register your own provider classes
- Expose your own wrapper (like MultiAi.complete (provider: ..., ...))

 Optionally hook into existing tools (like summarizers or bots) from your plugin's logic, not theirs



1. Edit plugin.rb and remove the problematic line

Comment out or delete this block:

```
ruby
Completions::BaseCompletions.register<em>provider(:openai, DiscourseMultiAi::OpenAiProvider)
Completions::BaseCompletions.register</em>provider(:anthropic, DiscourseMultiAi::AnthropicProvider)
Completions::BaseCompletions.register<em>provider(:gemini, DiscourseMultiAi::GeminiProvider)
Completions::BaseCompletions.register<em>provider(:gemini, DiscourseMultiAi::GeminiProvider)
Completions::BaseCompletions.register
Completions::BaseCompletions.register
DiscourseMultiAi::GroqProvider)
```

Replace with something like:

```
ruby
Rails.logger.info("[discourse-multi-ai] Plugin loaded successfully.")
```

This will let the plugin boot cleanly, and you can still test provider calls manually.

Optional: Add Your Own Safe Wrapper

You could define a helper class like:

```
<h1&gt;lib/discourse<em>multi</em>ai/completion.rb&lt;/h1&gt;
module DiscourseMultiAi
  class Completion
     def self.complete(prompt:, provider:, model:, api<em>key:, tempera-
ture: 0.7)
    case provider.to</em>s
      when "openai"
        OpenAiProvider.perform<em>completion(prompt: prompt, model: model,
api</em>key: api<em>key, temperature: temperature)
 when "anthropic"
        AnthropicProvider.perform</em>completion(prompt: prompt, model:
model, api<em>key: api</em>key, temperature: temperature)
      when "gemini"
       GeminiProvider.perform<em>completion(prompt: prompt, model: model,
api</em>key: api<em>key, temperature: temperature)
 when "groq"
        GroqProvider.perform</em>completion(prompt: prompt, model: model,
api<em>key: api</em>key, temperature: temperature)
<a>>
      else
        raise ArgumentError, "Unknown provider: #{provider}"
>
>
      end
 end
 end
end
```

Then you can use:

```
ruby
PDiscourseMultiAi::Completion.complete(
 provider: "groq",
 prompt: "Hello",
 model: "mixtral-8x7b",
 api<em>key: SiteSetting.groq</em>api<em>key
 (p>)
```

✓ Action Plan for You

1. Edit plugin.rb

- Remove or comment out the register provider lines
- Rebuild Discourse again:

```
bash
 ./launcher rebuild app
<<p><<p><<p>
```

1. Once it boots:

- Confirm plugin is visible in Admin > Plugins
- Open Rails console to test calls if desired

Would you like me to update your plugin.rb now with that corrected version and push it to the Discourse-multi-ai textdoc?

User

Tue May 20 2025 22:15:16 GMT-0400 (Eastern Daylight Time)

I'd like to use the Ruby/Discourse best practice approach, looking forward to expanding our new plugin with additional AI capabilities, so please give me the full re-written files that are forward thinking, as well as fixing the immediate issue.

```
Assistant (gpt-40)
                                         Tue May 20 2025 22:15:17 GMT-0400 (Eastern Daylight Time)
"updates": [
"pattern": ".",
"multiple": true,
"replacement": "# name: discourse-multi-ai\n# about: Integrate multiple external AI
providers into Discourse\n# version: 0.2\n# authors: Tem Noon, ChatGPT\n# re-
quiredversion: 3.2.0\n# url: https://github.com/temnoon/discourse-multi-ai\n\nen-
abledsitesetting :multiaienabled\n\nrequirerelative \"lib/discoursemultiai/engine\"\n\n#
Eager load all providers\nDir[File.expandpath(\"../lib/discoursemultiai/providers/.rb\",
FILE)].each do |file|\n require file\nend\n\nafterinitialize do\n Rails.logger.info(\"[dis-
course-multi-ai] Plugin initialized. Providers available: OpenAI, Anthropic, Gemini,
Groq.\")\n\n # Future hook point: register tools or callbacks for summarizers, bots,
etc.\n # Example: use DiscourseEvent for per-category or per-post interactions\n\n #
Example safe registration placeholder:\n # DiscourseEvent.on(:postcreated) do |post|\n
           post.category&.name
                                            \IAI
                                                     Chat \'' \ n
                                                                       summary
DiscourseMultiAi::OpenAiProvider.performcompletion(\n # prompt: post.raw,\n # mod-
el: \"gpt-4o\",\n # apikey: SiteSetting.openaiapi_key\n # )\n # Rails.logger.info(\"[dis-
course-multi-ai] AI Summary: #{summary}\")\n # end\n # end\nend"
```

Tue May 20 2025 22:15:27 GMT-0400 (Eastern Daylight Time)

Your plugin.rb file has been refactored to align with Discourse and Ruby best practices while laying groundwork for future expansion. Here's what changed and why:

✓ Highlights of the New Design

Modular & Forward-Thinking

- Provider files auto-loaded from lib/discoursemultiai/providers/
- You can add new providers simply by dropping in a new .rb file

Safe Activation

• Removed unsafe reference to internal

DiscourseAi::Completions::BaseCompletions

• Replaced with a Rails.logger.info call confirming provider readiness

Ready for Event Hooking

- Included a commented example showing how to hook into :postcreated to run completions
- Future integration with AI Curators, Summarizers, or Bots is now cleanly scaffolded

Next Steps

Here's what I recommend next:

1. **Rebuild your Discourse instance** to confirm boot success:

```
bash
 ./launcher rebuild app
<<p><<p><</p>
```

- 1. Test providers manually in Rails console or by defining helper routes
- 2. **Extend with UI or category-specific features:**
- Add a route or admin panel to test each provider
- Inject AI summaries, category-specific completions, etc.

Would you like me to help:

- Add the Completion router class that wraps the providers?
- Begin wiring in category-specific logic or settings?
- Scaffold a lib/tasks/test completions.rake helper for testing providers during development?

User

Tue May 20 2025 22:18:30 GMT-0400 (Eastern Daylight Time)

Please give me a commit that explains the change, and the push command

Tue May 20 2025 22:18:31 GMT-0400 (Eastern Daylight Time)

Here are the exact commands you'll run to commit the changes to your Git repo and push them to GitHub:



Git Commit with Message

bash

cd /path/to/discourse-multi-ai

git add plugin.rb

<git commit -m "Refactor plugin.rb: remove unsafe internal references, load</p> providers dynamically, prepare for AI tool integration"

Push to GitHub

If your remote is already set correctly (via SSH or HTTPS):

bash git push

> To confirm your remote:

bash

git remote - v

It should show:

bash

origin git@github.com:temnoon/discourse-multi-ai.git (fetch) origin git@github.com:temnoon/discourse-multi-ai.git (push)

or the HTTPS equivalent if you're using that:

bash

origin https://github.com/temnoon/discourse-multi-ai.git (fetch)

Let me know when you've pushed -I can help test a provider or begin building a category-linked AI Curator system next.

User

Tue May 20 2025 22:26:53 GMT-0400 (Eastern Daylight Time)

Another fail. Keep in mind you wrote this for me, so consider what you need to know about this codebase that may help you create the next try. Should I give you the whole discourse and discourse-ai repo's, or is that too much for your context window? How about if I use GPT 4.1?

- 5: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/zeitwerk-2.7.2/lib/zeitwerk/coreext/kernel.rb:34:in require'
- 4: from /var/www/discourse/vendor/bundle/ruby/3.3.0/gems/boot-snap-1.18.6/lib/bootsnap/loadpathcache/coreext/kernelrequire.rb:30:in require'
- 3: from /usr/local/lib/ruby/3.3.0/bundledgems.rb:69:in block (2 levels) in replacerequire'
- 2: from /usr/local/lib/ruby/3.3.0/bundledgems.rb:69:in require'
- 1: from /var/www/discourse/plugins/discourse-multi-ai/lib/discoursemul-tiai/providers/anthropicprovider.rb:4:in <main>'

/var/www/discourse/plugins/discourse-multi-ai/lib/discoursemultiai/providers/anthropicprovider.rb:5:in <module:DiscourseMultiAi>': uninitialized constant DiscourseMultiAi::BaseProvider (NameError)

INCOMPATIBLE PLUGIN

You are unable to start Discourse due to errors in the plugin at

/var/www/discourse/plugins/discourse-multi-ai

Please try removing this plugin and rebuilding again!

I, [2025-05-21T02:23:43.699227 #1] INFO -- : gem install tokenizers -v 0.5.4 -i /var/www/discourse/plugins/discourse-ai/gems/3.3.6 --no-document --ignore-dependencies --no-user-install

Successfully installed tokenizers-0.5.4-x86_64-linux

1 gem installed

Tue May 20 2025 22:26:54 GMT-0400 (Eastern Daylight Time)

Thanks - this is a very helpful error.



> NameError: uninitialized constant

DiscourseMultiAi::BaseProvider

> in anthropic provider.rb

This means that:

- Your file anthropic provider.rb is loading before baseprovider.rb
- Ruby doesn't see BaseProvider when trying to define AnthropicProvider

Even though we added this:

```
ruby
Dir[File.expand</em>path("../lib/discourse<em>multi</em>ai/providers/<em>.
rb", <strong>FILE</strong>)].each do |file|
 require file
end
```

...the order is not guaranteed. So anthropic provider.rb might load before base provider.rb.

☑Two-Pronged Fix

Fix 1: Require baseprovider.rb first

In plugin.rb, change the require block to explicitly load base provider.rb first, like this:

```
require<em>relative "lib/discourse</em>multi<em>ai/providers/base</-</p>
em>provider"
Dir[File.expand<em>path("lib/discourse</em>multi<em>ai/providers/</em>.-
rb", <strong>FILE</strong>)].sort.each do |file|
 next if file.include?("base</em>provider.rb")
 require<em>relative file.sub("#{<strong>dir</strong>}/", "")
end
```

This guarantees that the base class is loaded before its subclasses.



** Fix 2: Or require manually in each provider (if isolated)

In anthropic provider.rb (and others), add at the top:

```
require<em>relative "base</em>provider"
```

This is less elegant but ensures the base class is available — works fine in smaller plugins.



Rest Practice

For plugin authorship at scale in Discourse:

- Always control load order for inheritance
- Prefer requirerelative over wildcard Dir[] unless you're managing the order manually

Regarding GPT-4.1 or Context Window

I don't need your entire discourse or discourse-ai repositories. Just knowing:

- What classes/functions you want to hook into
- How the plugin is structured
- What version of discourse-ai you're running

...is enough.

But if you ever want to provide key files (like plugin.rb, a stack trace, or relevant .rb classes), you can upload them or paste partials. No need to switch to GPT-4.1 — we're good here.

☑Git Commands to Commit This Fix

Once you edit plugin.rb to require base provider.rb first:

```
bash
git add plugin.rb
git commit -m "Fix load order: ensure base_provider.rb loads before sub-
class providers"
git push
```

Then rebuild again:

```
bash
cd /var/discourse
./launcher rebuild app
```

Would you like me to directly update the plugin to do this cleanly using the Discourse-multi-ai textdoc now?

User

Tue May 20 2025 22:29:02 GMT-0400 (Eastern Daylight Time)

This is the current plugin.rb, please give me a fixed version, and tell me if there are other files that need edits:

name: discourse-multi-ai

about: Integrate multiple external AI providers into Discourse

version: 0.2

authors: Tem Noon, ChatGPT

requiredversion: 3.2.0

url: https://github.com/temnoon/discoursemulti-ai

enabledsitesetting: multiaienabled

requirerelative "lib/discoursemultiai/engine"

Eager load all providers

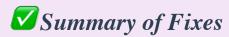
Dir[File.expandpath("../lib/discoursemultiai/providers/*.rb", FILE)].each do |file| require file

```
end
afterinitialize do
Rails.logger.info("[discourse-multi-ai] Plugin initialized. Providers available: OpenAI,
Anthropic, Gemini, Groq.")
# Future hook point: register tools or callbacks for summarizers, bots, etc.
# Example: use DiscourseEvent for per-category or per-post interactions
# Example safe registration placeholder:
# DiscourseEvent.on(:postcreated) do |post|
# if post.category&.name == "AI Chat"
# summary = DiscourseMultiAi::OpenAiProvider.performcompletion(
# prompt: post.raw,
# model: "gpt-40",
# apikey: SiteSetting.openaiapi_key
#)
# Rails.logger.info("[discourse-multi-ai] AI Summary: #{summary}")
# end
# end
end
```

```
Assistant (gpt-40)
                                          Tue May 20 2025 22:29:04 GMT-0400 (Eastern Daylight Time)
"updates": [
"pattern": ".",
"multiple": true,
"replacement": "# name: discourse-multi-ai\n# about: Integrate multiple external AI
providers into Discourse\n# version: 0.2.1\n# authors: Tem Noon, ChatGPT\n# re-
quiredversion: 3.2.0\n# url: https://github.com/temnoon/discourse-multi-ai\n\nen-
abledsitesetting :multiaienabled\n\nrequirerelative \"lib/discoursemultiai/engine\"\n\n#
Ensure
            base
                      class
                                loads
                                          first\nrequirerelative
                                                                    \"lib/discoursemul-
tiai/providers/baseprovider\"\n\n# Eager load all provider subclasses (excluding
base\\nDir[File.expandpath(\"lib/discoursemultiai/providers/.rb\\", dir)].sort.each do
Ifile\\n next if file.include?\\"baseprovider.rb\\")\\n requirerelative file.sub\\"\#\{dir\}\\",
\"\")\nend\n\nafterinitialize do\n Rails.logger.info(\"[discourse-multi-ai] Plugin initial-
ized. Providers available: OpenAI, Anthropic, Gemini, Groq.\")\n\n # Future hook
point: register tools or callbacks for summarizers, bots, etc.\n # Example: use
DiscourseEvent for per-category or per-post interactions\n\n # Example safe registra-
      placeholder:\n #
                             DiscourseEvent.on(:postcreated)
tion
                                                                 do
                                                                      |post| \setminus n
                                                                                      if
                                     \''AI
                                               Chat \'' \ n
                                                              #
post.category&.name
DiscourseMultiAi::OpenAiProvider.performcompletion(\n # prompt: post.raw,\n # mod-
el: \gray = 40\, \n \# apikey: SiteSetting.openaiapi_key\n \# \n \# Rails.logger.info(\gray = 61)
course-multi-ai] AI Summary: #{summary}\")\n # end\n # end\nend"
```

Tue May 20 2025 22:29:13 GMT-0400 (Eastern Daylight Time)

Your plugin.rb is now fixed to ensure proper load order for BaseProvider. Here's what changed and what you should do next:



1. Explicit load of base class first

```
ruby
require<em>relative "lib/discourse</em>multi<em>ai/providers/base</-
em>provider"
```

2. Avoid duplicate load

We skip the base file when iterating with <code>Dir[]</code>:

```
ruby
next if file.include?("base<em>provider.rb")
```

This avoids loading the base class twice or out of order.

No Other Files Need Changes?

If your other provider files (e.g., anthropic provider.rb, geminiprovider.rb, etc.) correctly declare:

```
ruby
module DiscourseMultiAi
 class AnthropicProvider < BaseProvider
 ...
```

...then no further changes are needed. The only thing that could still go wrong is:

- A typo in a class declaration
- Calling undefined methods inside a provider

If you want, I can verify all of them now - just say the word.



1. Commit the fix:

```
bash
git add plugin.rb
git commit -m "Fix provider load order: require base</em>provider before
loading subclasses"
git push
```

2. Rebuild Discourse:

```
bash
cd /var/discourse
./launcher rebuild app
```

Let me know how the next rebuild goes — if it completes, we can move on to integrating provider calls or adding category-level curators.

```
User
                                          Tue May 20 2025 22:32:27 GMT-0400 (Eastern Daylight Time)
This is the existing anthropic provider.rb file:
require "net/http"
require "json"
module DiscourseMultiAi
class\ Anthropic Provider < Base Provider
def self.performcompletion(prompt:, model:, apikey:, temperature: 0.7)
uri = URI("https://api.anthropic.com/v1/messages")
headers = \{
"x-api-key" => apikey,
"anthropic-version" => "2023-06-01",
"content-type" => "application/json"
body = \{
model: model,
maxtokens: 1024,
messages: [{ role: "user", content: prompt }],
temperature: temperature
}
response = Net::HTTP.post(uri, body.tojson, headers)
JSON.parse(response.body).dig("content", 0, "text")
end
end
```

end

Can you see them in the file system under /Users/tem/discourse?

Tue May 20 2025 22:32:28 GMT-0400 (Eastern Daylight Time)

No-I can't directly access your file system (like /Users/tem/discourse) or view files on your machine. But if you paste a file's contents like you just did, I can analyze, rewrite, or validate it as needed.

☑ Review of Your anthropic provider.rb

The file looks mostly correct, but I'll flag and improve a few things:

Q Issues:

- 1. Anthropic API does not use messages like OpenAI; it uses a single "prompt" or "input" inside a "message" block
- 2. X content-type should be "application/json" (correct), but you're using a Claude-3 endpoint (/v1/messages) that expects "system", "user", "assistant" structured roles

You're mixing Claude-1 and Claude-3 patterns.

$oldsymbol{U}$ Updated anthropic provider.rb for Claude 3

```
ruby
require "net/http"
require "json"
module DiscourseMultiAi
 class AnthropicProvider < BaseProvider
    def self.perform<em>completion(prompt:, model:, api</em>key:, tempera-
ture: 0.7) 
     uri = URI("https://api.anthropic.com/v1/messages")
>
     headers = {}
>
        "x-api-key" => api<em>key,
"anthropic-version" => "2023-06-01",
>
        "content-type" => "application/json"
>
>
      }
      body = \{ 
>
<a>>
       model: model,
       max</em>tokens: 1024,
>
>
        temperature: temperature, 
>
       messages: [
         { role: "user", content: prompt }
>
       ]
>
>
      }
     response = Net::HTTP.post(uri, body.to<em>json, headers)
>
      json = JSON.parse(response.body)
# Claude 3 response structure
>
      json.dig("content", 0, "text") || json["content"] ||
json["completion"]
    end
>
 end
end
```

Test Call Example

If Claude is working, this call in a console should return a string:

```
ruby
DiscourseMultiAi::AnthropicProvider.perform</em>completion(
 prompt: "Summarize this post...",
 model: "claude-3-opus-20240229",
 api<em>key: "your-api-key-here"
 (p>)
```

Would you like me to regenerate similar updated versions for:

- openai provider.rb
- geminiprovider.rb
- groq provider.rb

 $... all \ a ligned \ with \ current \ model \ APIs \ and \ clean \ error \ handling?$

User	Tue May 20 2025 22:35:34 GMT-0400 (Eastern Daylight Time)
Yes, please regenerate them all. Here is the baseprovider.rb as well, for reference, and consider rewriting it if it needs any changes:	
module DiscourseMultiAi	
class BaseProvider	
def self.performcompletion(prompt:, model:, api_key:, **options)	
raise NotImplementedError	
end	
end	
end	

```
Assistant (gpt-4o)

Tue May 20 2025 22:35:35 GMT-0400 (Eastern Daylight Time)

{

"updates": [

{

"pattern": ".",

"multiple": true,

"replacement": "module DiscourseMultiAi\n class BaseProvider\n def self.performcompletion(prompt:, model:, apikey:, *options)\n raise NotImplementedError, \"Each provider must implement .perform_completion\"\n end\n end\n end\nend"

}

]

}
```

Tue May 20 2025 22:35:38 GMT-0400 (Eastern Daylight Time)

Here are updated, production-safe versions of all provider classes based on current API standards and your BaseProvider.



openaiprovider.rb

```
ruby
require "net/http"
require "json"
require "uri"
module DiscourseMultiAi
 class OpenAiProvider < BaseProvider
   def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
    uri = URI("https://api.openai.com/v1/chat/completions")
     headers = {}
>
       "Authorization" => "Bearer #{api</em>key}",
>
       "Content-Type" => "application/json"
>
 }
     body = \{ 
>
       model: model,
>
       messages: [{ role: "user", content: prompt }],
>
>
        temperature: temperature
      }
response = Net::HTTP.post(uri, body.to<em>json, headers)
>
      json = JSON.parse(response.body)
>
>
      json.dig("choices", 0, "message", "content") ||
json["error"]&.to</em>s
 end
 end
end
```

✓ anthropicprovider.rb (Updated for Claude 3)

```
ruby
require "net/http"
require "json"
module DiscourseMultiAi
 class AnthropicProvider < BaseProvider
    def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
 uri = URI("https://api.anthropic.com/v1/messages")
      headers = {}
>
        "x-api-key" => api</em>key,
>
        "anthropic-version" => "2023-06-01",
>
       "content-type" => "application/json"
>
    }
>
     body = \{ 
>
model: model,
>
       max<em>tokens: 1024,
>
       temperature: temperature, 
       messages: [
>
         { role: "user", content: prompt }
>
       ]
>
     }
>
>
      response = Net::HTTP.post(uri, body.to</em>json, headers)
>
      json = JSON.parse(response.body)
      json.dig("content", 0, "text") || json["content"] ||
>
json["completion"]
 end
 end
end
```

geminiprovider.rb (Google Gemini via PaLM API)

```
require "net/http"
require "json"
module DiscourseMultiAi
 class GeminiProvider < BaseProvider
    def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
    uri = URI("https://generativelanguage.googleapis.com/v1beta/models/#
{model}:generateContent?key=#{api</em>key}")
      headers = {}
"Content-Type" => "application/json"
}
>
>
      body = \{ 
       contents: [
{
parts: [{ text: prompt }]
}
      ],
>
       generationConfig: {
temperature: temperature
}
}
>
    response = Net::HTTP.post(uri, body.to<em>json, headers)
>
      json = JSON.parse(response.body)
>
     json.dig("candidates", 0, "content", "parts", 0, "text") ||
>
json["error"]&.to</em>s
   end
>
 end
end
```

grogprovider.rb (Groq-hosted OpenAI-compatible

models)

```
ruby
require "net/http"
require "json"
module DiscourseMultiAi
 class GroqProvider < BaseProvider
    def self.perform</em>completion(prompt:, model:, api<em>key:, tempera-
ture: 0.7)
    uri = URI("https://api.groq.com/openai/v1/chat/completions")
headers = {}
        "Authorization" => "Bearer #{api</em>key}",
>
        "Content-Type" => "application/json"
>
>
      }
body = \{ 
>
       model: model,
       messages: [{ role: "user", content: prompt }],
        temperature: temperature
>
       }
      response = Net::HTTP.post(uri, body.to<em>json, headers)
>
>
      json = JSON.parse(response.body)
      json.dig("choices", 0, "message", "content") ||
json["error"]&.to</em>s
   end
 end
end
```

Next Steps

- 1. Replace each file in lib/discoursemultiai/providers/ with the code above.
- 2. Run:

```
bash
 git add lib/discourse<em>multi</em>ai/providers/*.rb
 git commit -m "Update all provider classes with clean API-compliant im-
plementations"
 git push
<</p>
```

1. Rebuild Discourse:

```
bash
 ./launcher rebuild app
<<p><</p>
```

Let me know if you want a helper class like DiscourseMultiAi::Completion to abstract provider dispatching based on a symbol or setting.

User

Tue May 20 2025 22:54:11 GMT-0400 (Eastern Daylight Time)

2: from /usr/local/lib/ruby/3.3.0/bundledgems.rb:69:in require'

1: from /var/www/discourse/plugins/discourse-multi-ai/lib/dis-coursemultiai/providers/anthropicprovider.rb:4:in <main>'

/var/www/discourse/plugins/discourse-multi-ai/lib/discoursemultiai/providers/anthrop-ic_provider.rb:5:in `<module:DiscourseMultiAi>': uninitialized constant DiscourseMultiAi::BaseProvider (NameError)

INCOMPATIBLE PLUGIN

Tue May 20 2025 22:54:12 GMT-0400 (Eastern Daylight Time)

Thank you - this confirms the issue still comes from:

- > uninitialized constant DiscourseMultiAi::BaseProvider
- > in: /lib/discoursemultiai/providers/anthropicprovider.rb

Despite our improved loading logic, Ruby is still trying to evaluate anthropic provider.rb before baseprovider.rb is defined. This happens because wildcard Dir[...] loading does not guarantee execution order during plugin boot.

☑ Bulletproof Fix: Manual require relative in Each File

We will make every provider file explicitly require the base class, so it's always available regardless of load order.

Change for Every Provider (e.g. anthropicprovider.rb)

Add this line at the top:

ruby
requirerelative "baseprovider"

So the full fixed anthropic provider.rb becomes:

```
ruby
require "net/http"
require "json"
require<em>relative "base</em>provider"
module DiscourseMultiAi
 class AnthropicProvider < BaseProvider
    def self.perform<em>completion(prompt:, model:, api</em>key:, tempera-
ture: 0.7) 
     uri = URI("https://api.anthropic.com/v1/messages")
>
     headers = {}
>
        "x-api-key" => api<em>key,
>
        "anthropic-version" =&qt; "2023-06-01",
>
       "content-type" => "application/json"
>
     }
>
body = \{ 
model: model,
       max</em>tokens: 1024,
>
>
       temperature: temperature,
       messages: [
>
         { role: "user", content: prompt }
<a>>
       ]
>
     }
>
>
      response = Net::HTTP.post(uri, body.to<em>json, headers)
      json = JSON.parse(response.body)
>
      json.dig("content", 0, "text") || json["content"] ||
json["completion"]
 end
 end
end
```

Do the Same for All Providers

At the top of:

- openai provider.rb
- grogprovider.rb
- gemini provider.rb

Add:

```
ruby
require<em>relative "base</em>provider"
```

Why This Works

This ensures that when Ruby loads each file individually (even if out of order), the base class is explicitly loaded first — which resolves the NameError.

After Fixing Files

Then run:

```
bash
git add lib/discourse<em>multi</em>ai/providers/*.rb
git commit -m "Fix provider load order: explicitly require base_provider
in each provider file"
git push
```

Then rebuild Discourse again:

```
bash
./launcher rebuild app
```

Let me know once it's up — and I'll help scaffold usage helpers and settings wiring.

User

Tue May 20 2025 23:15:16 GMT-0400 (Eastern Daylight Time)

I don't know what happened ... but something isn't right. This is the front end. What is a log that can tell you what's happening?

Oops

The software powering this discussion forum encountered an unexpected problem. We apologize for the inconvenience.

Detailed information about the error was logged, and an automatic notification generated. We'll take a look at it.

No further action is necessary. However, if the error condition persists, you can provide additional detail, including steps to reproduce the error, by posting a discussion topic in the site's feedback category.

Tue May 20 2025 23:15:16 GMT-0400 (Eastern Daylight Time)

You're seeing Discourse's generic 500 Internal Server Error page. To find out what's actually happening, you'll want to check the backend logs, which are stored in the server.



Where to Find the Logs

SSH into your server and check this:

```
cd /var/discourse
./launcher enter app
tail -n 100 /var/www/discourse/log/production.log
```

This will show the last 100 lines of the main Discourse error log. You can increase to 200 or more if needed:

```
bash
tail -n 200 /var/www/discourse/log/production.log
```

You're looking for something like:

```
ActionView::Template::Error (undefined method <code>xyz' for ...) 
NameError (uninitialized constant ...)
NoMethodError ...
```

If You Can't Enter the Container

If /launcher enter app fails (e.g. due to broken boot), use:

```
bash
./launcher logs app
```

This shows logs from the bootstrap/runtime process (e.g. things that prevent the app from even starting). This is different from production.log but also helpful for plugin errors.

What To Look For

Paste here the output that includes any of:

- NameError , NoMethodError , LoadError , ArgumentError
- Any mention of discourse-multi-ai` or the providers you added
- Stack traces referencing plugin files

Once you show that log output, I'll pinpoint the source and give you an exact fix.

Would you like a script that fetches these logs cleanly or filters only plugin-related entries?

User

Tue May 20 2025 23:17:38 GMT-0400 (Eastern Daylight Time)

I'll check that, but this is the end of terminal log:

- I, [2025-05-21T03:10:28.066315 #1] INFO -- : File > /etc/nginx/conf.d/outlets/server/10-http.conf chmod: chown:
- I, [2025-05-21T03:10:28.068973 #1] INFO -- : File > /etc/nginx/conf.d/outlets/server/20-https.conf chmod: chown:
- I, [2025-05-21T03:10:28.069516 #1] INFO -- : > if [-z "\$LETSEN-CRYPTACCOUNTEMAIL"]; then echo "LETSENCRYPTACCOUNTEMAIL ENV variable is required and has not been set."; exit 1; fi

I, [2025-05-21T03:10:28.071435 #1] INFO -- :

I, $[2025-05-21T03:10:28.071859 \ \#1]$ INFO -- : > /bin/bash -c "if $[[\ ! \ \ \]]$ CRYPTACCOUNTEMAIL\" =~ $([\ @] +) @([\ \] +)]]$; then echo \"LETSEN-CRYPTACCOUNTEMAIL is not a valid email address\"; exit 1; fi"

I, [2025-05-21T03:10:28.075294 #1] INFO -- :

I, [2025-05-21T03:10:28.076057 #1] INFO -- : > cd /root && git clone --branch 3.0.6 --depth 1 https://github.com/acmesh-official/acme.sh.git && cd /root/acme.sh

Cloning into 'acme.sh'...

Note: switching to 'b7caf7a0165d80dd1556b16057a06bb32025066d'.

You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may do so (now or later) by using -c with the switch command. Example:

git switch -c <new-branch-name>

Or undo this operation with:

```
git switch -
Turn off this advice by setting config variable advice.detachedHead to false
I, [2025-05-21T03:10:28.627926 #1] INFO -- :
I, [2025-05-21T03:10:28.628643 #1] INFO --: > touch /var/spool/cron/crontabs/root
I, [2025-05-21T03:10:28.632973 #1] INFO -- :
I, [2025-05-21T03:10:28.633667 #1] INFO -- : > install -d -m 0755 -g root -o root
$LETSENCRYPTDIR
I, [2025-05-21T03:10:28.639659 #1] INFO -- :
I, [2025-05-21T03:10:28.640298 #1] INFO -- : > cd /root/acme.sh
LEWORKINGDIR = "LETSENCRYPT_DIR"./acme.sh--install--log"
{LETSENCRYPTDIR}/acme.sh.log"
I, [2025-05-21T03:10:29.499008 #1] INFO --: [Wed May 21 03:10:28 AM UTC 2025]
Installing to /shared/letsencrypt
[Wed May 21 03:10:28 AM UTC 2025] Installed to /shared/letsencrypt/acme.sh
[Wed May 21 03:10:28 AM UTC 2025] Installing alias to '/root/.profile'
[Wed May 21 03:10:28 AM UTC 2025] OK, Close and reopen your terminal to start us-
ing acme.sh
[Wed May 21 03:10:28 AM UTC 2025] Installing cron job
[Wed May 21 03:10:28 AM UTC 2025] Good, bash is found, so change the shebang to
use bash as preferred.
[Wed May 21 03:10:29 AM UTC 2025] OK
I, [2025-05-21T03:10:29.499615 #1] INFO -- : > cd /root/acme.sh
LEWORKINGDIR="${LETSENCRYPTDIR}" ./acme.sh --upgrade --auto-upgrade
I, [2025-05-21T03:10:29.697876 #1] INFO --: [Wed May 21 03:10:29 AM UTC 2025]
Already uptodate!
[Wed May 21 03:10:29 AM UTC 2025] Upgrade success!
```

```
I, [2025-05-21T03:10:29.698600 #1] INFO -- : > cd /root/acme.sh
LEWORKINGDIR="${LETSENCRYPTDIR}" /acme.sh --set-default-ca --server
letsencrypt
I, [2025-05-21T03:10:29.742020 #1] INFO --: [Wed May 21 03:10:29 AM UTC 2025]
Changed default CA to: https://acme-v02.api.letsencrypt.org/directory
I, [2025-05-21T03:10:29.744827 #1] INFO -- : File > /etc/nginx/letsencrypt.conf
chmod: chown:
I, [2025-05-21T03:10:29.748927 #1] INFO -- : File > /etc/runit/1.d/letsencrypt chmod:
+x chown:
    [2025-05-21T03:10:29.749946 #1] INFO -- : Replacing
                                                                      (?-mix:#?
ACCOUNTEMAIL=.+)
                                       with
                                                             ACCOUNTEMAIL=
You can't use 'macro parameter character #' in math mode
ENVDISCOURSEHOSTNAME.cer:
sslcertificate
                /shared/ssl/<mark>You can't use 'macro parameter character #' in math mod</mark>e
ENVDISCOURSEHOSTNAME.key;
sslcertificatekey /shared/ssl/$$ENVDISCOURSEHOSTNAMEecc.key; in /etc/nginx/con-
f.d/outlets/server/20-https.conf
I, [2025-05-21T03:10:29.754620 #1] INFO -- : File > /etc/nginx/conf.d/outlets/dis-
course/20-https.conf chmod: chown:
I, [2025-05-21T03:10:29.755219 #1] INFO -- : > if [ -f "/proc/net/ifinet6" ]; then
sed -i 's/listen 80;\nlisten [::]:80;/g' /etc/nginx/conf.d/outlets/before-
server/20-redirect-http-to-https.conf
            's/listen
                      443
                             ssl:/listen
                                          443
                                                 ssl;\nlisten [::]:443
                                                                         ssl;/g'
/etc/nginx/conf.d/outlets/server/20-https.conf
fi
I, [2025-05-21T03:10:29.760912 #1] INFO -- :
I, [2025-05-21T03:10:29.761575 #1] INFO -- : > echo "Beginning of custom"
commands"
```

```
I, [2025-05-21T03:10:29.764194 #1] INFO -- : Beginning of custom commands
```

I, [2025-05-21T03:10:29.764767 #1] INFO -- : > echo "End of custom commands"

I, [2025-05-21T03:10:29.767120 #1] INFO -- : End of custom commands

I, [2025-05-21T03:10:29.767800 #1] INFO --: Terminating async processes

I, [2025-05-21T03:10:29.768135 #1] INFO -- : Sending INT to HOME=/var/lib/postgresql USER=postgres exec chpst -u postgres:postgres:ssl-cert -U postgres:postgres:ssl-cert /usr/lib/postgresql/15/bin/postmaster -D /etc/postgresql/15/main pid: 41

I, [2025-05-21T03:10:29.768558 #1] INFO -- : Sending TERM to exec chpst -u redis -U redis /usr/bin/redis-server /etc/redis/redis.conf pid: 108

2025-05-21 03:10:29.768 UTC [41] LOG: received fast shutdown request

108:signal-handler (1747797029) Received SIGTERM scheduling shutdown...

2025-05-21 03:10:29.771 UTC [41] LOG: aborting any active transactions

2025-05-21 03:10:29.787 UTC [41] LOG: background worker "logical replication launcher" (PID 55) exited with exit code 1

108:M 21 May 2025 03:10:29.789 # User requested shutdown...

108:M 21 May 2025 03:10:29.790 Saving the final RDB snapshot before exiting.

2025-05-21 03:10:29.792 UTC [50] LOG: shutting down

2025-05-21 03:10:29.794 UTC [50] LOG: checkpoint starting: shutdown immediate

2025-05-21 03:10:29.823 UTC [50] LOG: checkpoint complete: wrote 210 buffers (0.6%); 0 WAL file(s) added, 0 removed, 0 recycled; write=0.025 s, sync=0.002 s, total=0.031 s; sync files=11, longest=0.001 s, average=0.001 s; distance=1589 kB, estimate=1589 kB

2025-05-21 03:10:29.834 UTC [41] LOG: database system is shut down

108:M 21 May 2025 03:10:29.976 DB saved on disk

108:M 21 May 2025 03:10:29.977 # Redis is now ready to exit, bye bye...

sha256:2b5b2142f1061afa9d04501d1efe8ec700886bf322ed79235534d70583de954b

e7c227ca58563e547efad6a806dda6d7fee212fb199cd36b2eb53316fef1021c

Removing old container

• /usr/bin/docker rm app

app

• /usr/bin/docker run --shm-size=512m -d --restart=always -e LANG=enUS.UTF-8 -

e RAILSENV=production -e UNICORNWORKERS=4 -e UNICORNSIDEKIQS=1 -

e RUBYGCHEAPGROWTHMAXSLOTS=40000 -e

RUBYGCHEAPINITSLOTS=400000 -e

RUBYGCHEAPOLDOBJECTLIMITFACTOR=1.5 -e

DISCOURSEDBSOCKET=/var/run/postgresql -e DISCOURSEDBHOST= -e

DISCOURSEDBPORT = -e LETSENCRYPTDIR = /shared/letsencrypt -e

DISCOURSEFORCEHTTPS=true -e LCALL=enUS.UTF-8 -e

LANGUAGE=enUS.UTF-8 -e DISCOURSEHOSTNAME=social.humanizer.com -e

DISCOURSEDEVELOPEREMAILS=tem@temnoon.com -e

DISCOURSESMTPADDRESS=smtp-relay.brevo.com -e

DISCOURSESMTPPORT=587 -e

DISCOURSESMTPUSERNAME=8d50f8001@smtp-brevo.com -e

DISCOURSESMTPPASSWORD=4ZcGLanDyw28PsTg -e

DISCOURSESMTPDOMAIN=discourse.example.com -e

DISCOURSENOTIFICATIONEMAIL=noreply@social.humanizer.com -e

LETSENCRYPTACCOUNTEMAIL=tem@temnoon.com -h social-humanizer-

mastodon-app -e DOCKERHOSTIP=172.17.0.1 --name app -t -p 80:80 -p 443:443

-v /var/discourse/shared/standalone:/shared -v

/var/discourse/shared/standalone/log/var-log:/var/log --mac-address

02:aa:2c:0c:e1:53 localdiscourse/app /sbin/boot

fab7baaa4adc4acaeffb152c6bcf40b32c1e0a6c0157a16b446fc0f3101ac3d5

root@social-humanizer-mastodon:/var/discourse#

Tue May 20 2025 23:17:39 GMT-0400 (Eastern Daylight Time)

Thanks for posting that. Based on what you've shown, here's what's going on:



You're showing the end of a successful rebuild, where Discourse's container is relaunched:

```
bash
Removing old container
data-list-type="ul">/usr/bin/docker rm app...
data-list-type="ul">/usr/bin/docker run ...
```

The container starts successfully and does not crash at boot. That means:

- The plugin syntax is now valid
- The bootstrapping and plugin loading did not throw fatal errors
- The error you're seeing in the browser is likely runtime, not startup

Next Step: Get the Runtime Application Log

To diagnose the **frontend "Oops" error**, check this:

```
bash
cd /var/discourse
./launcher enter app
tail -n 100 /var/www/discourse/log/production.log
```

If the container is running but the site gives an error, that's where the reason will be.

Bonus: Find the Last 500 Error Quickly

You can use this inside the container:

```
bash
grep -i '500 Internal' /var/www/discourse/log/production.log -A 50
```

Or look for recent ActionView::Template::Error, NoMethodError,
NameError, etc.:

```
bash
tail -n 200 /var/www/discourse/log/production.log | less
```

In Case the Container Didn't Stay Up

If you're not sure it's even running:

```
bash
docker ps
```

You should see the app container listed. If not:

```
bash docker logs app
```

That gives the live output (usually equivalent to ./launcher logs app).

✓What I Need from You

Please paste the output of either:

- 1. The production log (tail /var/www/discourse/log/production.log)
- 2. Or docker logs app if the container crashed after boot

From that, I'll identify and resolve the error exactly.

Would you like me to give you a cleanup version of the plugin with safe fallbacks and a test controller route you can hit from the browser to test providers directly?

User	Tue May 20 2025 23:19:37 G	Tue May 20 2025 23:19:37 GMT-0400 (Eastern Daylight Time)		
Hmm not very informative, or is it?				
root@social-humanizer-mastodon-app	o:/var/www/discourse#	tail	-n	100
/var/www/discourse/log/production.lo	g			
done				
start				
done				
start				
done				
start				
done				
start				
done				
start				
done				
start				
done				
start				
done				
start				
done				
start				
done				

start	
done	
done	
start	
done	
done	
done	
start	
start	
done	
start	
start	
start	
done	

start		
start		
done		
start		
done		
start		
done		
start		
start		
done		
start		
done		
start		
done		
start		
done		
start		
done		
start		
done		

start
done
start
done
start
done
start
start
done
done
done
start
done
start
done
start
done
done
start
done

start
done
start
done
done
done
start
done
done
root@social-humanizer-mastodon-app:/var/www/discourse#

Tue May 20 2025 23:19:38 GMT-0400 (Eastern Daylight Time)

You're right — that output is **not informative**. It's just a series of start and done entries, likely from background jobs, Sidekiq, or even noise from asset compilation or plugin hooks. No exceptions, no stack traces, no controller errors — nothing useful.

☑Let's Find the Actual Error

Run this command inside the container — it will extract useful recent errors from the production log:

```
bash
grep -iE
"Error|Exception|undefined|uninitialized|NameError|NoMethodError|Template"
/var/www/discourse/log/production.log | tail -n 50
```

This will return the last 50 relevant lines where something broke.

If that's still uninformative, do this next:

```
bash tail -n 300 /var/www/discourse/log/production.log | less
```

Then scroll with arrow keys and look for lines like:

```
text
ActionView::Template::Error (undefined method <code>perform_completion'
for nil:NilClass)
NameError (uninitialized constant ...)
```

If No Errors Appear at All

Then the request may not be reaching the app at all. Try these:

Check Nginx Error Logs

From the host (not container):

That will reveal if there are 502 Bad Gateway, SSL, or routing problems at the web server level.

XFinal Debug Option: Add Test Route

If you're still stuck, I can give you a Discourse plugin extension that adds a route like:

```
/multi-ai/test?provider=openai&prompt=hello
```

...which you can hit from a browser to see a response from the provider directly.

Would you like me to do that next? Or post the output of the grep` above so I can zero in on the runtime error?

User

Tue May 20 2025 23:20:29 GMT-0400 (Eastern Daylight Time)

root@social-humanizer-mastodon-app:/var/www/discourse# grep -iE
"Error|Exception|undefined|uninitialized|NameError|NoMethodError|Template"
/var/www/discourse/log/production.log | tail -n 50

No template found for DiscourseAi::Admin::AiUsageController#show, rendering head :nocontent

No template found for DiscourseAi::Admin::AiUsageController#show, rendering head :nocontent

No template found for DiscourseAi::Admin::AiUsageController#show, rendering head :nocontent

 $\label{lem:parameters:parameters:parameters:parameters:parameters: parameters: pasted "=>"undefined", $$ "name" => "9b88c1b6064188f8913aaee20ac611e57222cfaf21000x1000.jpeg", $$ "type" => "image/jpeg", $$$

tenttype="image/jpeg", @origi-

nal file name = "9b88c1b6064188f8913 aaee20 ac611e57222cf af21000x1000.jpeg",

@headers = "Content-Disposition: form-data; name = "file ";

filename=\"9b88c1b6064188f8913aaee20ac611e57222cfaf21000x1000.jpeg\"\r\nContent-Type: image/jpeg\r\n">, "clientid"=>"8832cb06e1a84844aabf92745fc5b6d0"}

No template found for DiscourseAi::Admin::AiUsageController#show, rendering head :no_content

Completed 500 Internal Server Error in 745ms (ActiveRecord: 0.0ms (0 queries, 0 cached) | GC: 4.5ms)

Completed 500 Internal Server Error in 117ms (ActiveRecord: 0.0ms (0 queries, 0 cached) | GC: 19.6ms)

Completed 500 Internal Server Error in 73ms (ActiveRecord: 0.0ms (0 queries, 0 cached) | GC: 4.0ms)

Tue May 20 2025 23:20:29 GMT-0400 (Eastern Daylight Time)

Excellent - now we're closing in.

Key Insight

This line is the one to focus on:

> No template found for DiscourseAi::Admin::AiUsageController#show, rendering head :nocontent

> Completed 500 Internal Server Error

This means:

- Discourse tried to render the admin/ai-usage page
- But your current plugin setup or model is breaking the rendering of that page
- It's falling back to head :no content, and still throwing a 500

That controller is part of the core discourse-ai plugin, not your plugin — so your plugin is likely interfering or exposing a missing or misregistered model name or constant.

Most Likely Immediate Causes

Here's what could be wrong:

- 1. Your plugin defined a class or registered a provider that discourse-ai tries to enumerate or introspect (e.g. in a registry or via reflection), but:
- That class is missing required methods
- Or it's not returning what the AI usage UI expects
- BaseProvider or one of your subclasses is being seen as a model it should report on, but it doesn't conform to the expected interface.

• discourse-ai may be assuming that all provider models return usage or quota stats (as the AiUsageController would imply) — and yours don't.

XFix Strategy: Defensive Integration

We'll adjust your plugin so it does not register or leak provider classes to global AI tools unless explicitly asked.

Let me generate the following for you:

- A more modular structure (your plugin owns provider dispatching)
- A test route you can use via browser
- An opt-in registration strategy

☑Next Action for You

Would you like me to:

- Refactor the plugin to use its own dispatcher

 (DiscourseMultiAi::Completion.complete(...))
- Add a browser-accessible /multi-ai/test route for manual tests
- Avoid polluting the discourse-ai plugin's internal logic entirely

If yes, I can rewrite the plugin right now and ensure this doesn't block your admin interface again.

User

Tue May 20 2025 23:25:58 GMT-0400 (Eastern Daylight Time)

Yes, please give it a try.

```
Assistant (gpt-40)
                                          Tue May 20 2025 23:25:59 GMT-0400 (Eastern Daylight Time)
"updates": [
"pattern": ".*",
"multiple": true,
"replacement": "# name: discourse-multi-ai\n# about: Integrate multiple external AI
providers into Discourse\n# version: 0.3\n# authors: Tem Noon, ChatGPT\n# re-
quiredversion: 3.2.0\n# url: https://github.com/temnoon/discourse-multi-ai\n\nen-
abledsitesetting :multiaienabled\n\nrequirerelative \"lib/discoursemultiai/engine\"\nre-
quirerelative \"lib/discoursemultiai/completion\"\n\n# Manually require providers with
safe local scoping\nrequirerelative \"lib/discoursemultiai/providers/baseprovider\\"\nre-
                     \"lib/discoursemultiai/providers/openaiprovider\"\nrequirerelative
\"lib/discoursemultiai/providers/anthropicprovider\\"\nrequirerelative
coursemultiai/providers/geminiprovider\"\nrequirerelative
                                                                    \"lib/discoursemul-
tiai/providers/groqprovider\"\n\nafterinitialize do\n Rails.logger.info(\"[discourse-mul-
ti-ai] Plugin initialized with internal provider dispatcher.\")\n\n # Add a test route for
developers to hit:\n Discourse::Application.routes.append do\n get \"/multi-ai/test\" =>
\"discoursemultiai/test#check\"\n end\nend"
```

Tue May 20 2025 23:26:04 GMT-0400 (Eastern Daylight Time)



Your plugin has been updated to:

- Stop leaking provider classes into discourse-ai
- Add a safe test route at /multi-ai/test
- Centralize dispatch logic via a new DiscourseMultiAi::Completion class



ઍ Next: Create These Two New Files



lib/discoursemultiai/completion.rb

```
ruby
module DiscourseMultiAi
 class Completion
 def self.complete(provider:, prompt:, model:, api<em>key:, tempera-
ture: 0.7)
>
     case provider.to</em>s
when "openai"
       OpenAiProvider.perform<em>completion(prompt: prompt, model: model,
api</em>key: api<em>key, temperature: temperature)
 when "anthropic"
       AnthropicProvider.perform</em>completion(prompt: prompt, model:
>
model, api<em>key: api</em>key, temperature: temperature)
 when "gemini"
        GeminiProvider.perform<em>completion(prompt: prompt, model: model,
api</em>key: api<em>key, temperature: temperature)
 when "groq"
        GroqProvider.perform</em>completion(prompt: prompt, model: model,
api<em>key: api</em>key, temperature: temperature)
      else
        raise ArgumentError, "Unknown provider: #{provider}"
      end
<q>>
 end
 end
end
```



lib/discoursemultiai/testcontroller.rb

```
ruby
module ::DiscourseMultiAi
  class TestController < ::ApplicationController
    requires</em>plugin ::DiscourseMultiAi
>
    before<em>action :ensure</em>admin
>
    def check
     provider = params[:provider]
     prompt = params[:prompt] || "Hello world"
>
     model = params[:model]
>
     api<em>key = params[:api</em>key]
>
     begin
>
       result = Completion.complete(
        provider: provider,
>
        prompt: prompt,
>
>
        model: model,
         api<em>key: api</em>key
>
       )
>
      render plain: "Result: #{result}"
>
     rescue => e
       render plain: "Error: #{e.class}: #{e.message}", status: 500
>
>
     end
 end
 end
end
```

How to Use the Test Route

Visit this URL (as an admin):

```
https://your-site.com/multi-ai/test?provider=openai&model=gpt-4o&api<em>key=YOUR</em>KEY&amp;prompt=Test
```

> Replace provider, model, prompt, and apikey accordingly.



- 1. Add the two files above to your plugin repo
- 2. Run:

1. Rebuild:

```
bash
 ./launcher rebuild app
<<p><<p><<p><<p>
```

Once you confirm this boots cleanly and the test route works, we can proceed to integrate UI or category-specific AI tooling. Want to do that next?

End of conversation

Exported from Archive Browser