Add model groq

 **File:** packages/data-provider/src/enums.ts

 **Change:** Add groq to the EModelEndpoint enum.

// packages/data-provider/src/enums.ts

export enum EModelEndpoint {

google = 'google',

openAI = 'openAI',

azureOpenAI = 'azureOpenAI',

bingAI = 'bingAI',

anthropic = 'anthropic',

chatGPTBrowser = 'chatGPTBrowser',

gptPlugins = 'gptPlugins',

lemur = 'lemur',

sdxl = 'sdxl',

groq = 'groq', // <-- Add this line

}

**2. Add an Icon for Groq**

To make your new endpoint look professional, you'll need an icon.

* **File:** Create a new file at client/src/components/svg/GroqIcon.tsx
* **Change:** Add the SVG for your new icon. Here is a sample:

// client/src/components/svg/GroqIcon.tsx

import React from 'react';

const GroqIcon = (props) => (

<svg

{...props}

xmlns="http://www.w3.org/2000/svg"

viewBox="0 0 24 24"

fill="none"

stroke="currentColor"

strokeWidth="2"

strokeLinecap="round"

strokeLinejoin="round"

>

<circle cx="12" cy="12" r="10" />

<path d="M8 12h8" />

<path d="M12 8v8" />

</svg>

);

export default GroqIcon;

 **File:** client/src/hooks/Endpoint/Icons.tsx

 **Change:** Import your new GroqIcon and add it to the endpointIcons object.

// client/src/hooks/Endpoint/Icons.tsx

//... other imports

import GroqIcon from '~/components/svg/GroqIcon';

//...

export const endpointIcons = {

//... other endpoints

[EModelEndpoint.groq]: GroqIcon, // <-- Add this line

};

Of course! To add a "Groq" model with a "Compound" sub-category and a settings button, you'll primarily be working within the client and packages directories. The key is to add the new endpoint and then create a special case for its message handling. Here are the files you'll need to modify and the changes to make:

**1. Define the New Endpoint**

You need to let the application know that "Groq" is a valid endpoint.

* **File:** packages/data-provider/src/enums.ts
* **Change:** Add groq to the EModelEndpoint enum.

TypeScript

// packages/data-provider/src/enums.ts

export enum EModelEndpoint {

google = 'google',

openAI = 'openAI',

azureOpenAI = 'azureOpenAI',

bingAI = 'bingAI',

anthropic = 'anthropic',

chatGPTBrowser = 'chatGPTBrowser',

gptPlugins = 'gptPlugins',

lemur = 'lemur',

sdxl = 'sdxl',

groq = 'groq', // <-- Add this line

}

**2. Add an Icon for Groq**

To make your new endpoint look professional, you'll need an icon.

* **File:** Create a new file at client/src/components/svg/GroqIcon.tsx
* **Change:** Add the SVG for your new icon. Here is a sample:

TypeScript

// client/src/components/svg/GroqIcon.tsx

import React from 'react';

const GroqIcon = (props) => (

<svg

{...props}

xmlns="http://www.w3.org/2000/svg"

viewBox="0 0 24 24"

fill="none"

stroke="currentColor"

strokeWidth="2"

strokeLinecap="round"

strokeLinejoin="round"

>

<circle cx="12" cy="12" r="10" />

<path d="M8 12h8" />

<path d="M12 8v8" />

</svg>

);

export default GroqIcon;

* **File:** client/src/hooks/Endpoint/Icons.tsx
* **Change:** Import your new GroqIcon and add it to the endpointIcons object.

TypeScript

// client/src/hooks/Endpoint/Icons.tsx

//... other imports

import GroqIcon from '~/components/svg/GroqIcon';

//...

export const endpointIcons = {

//... other endpoints

[EModelEndpoint.groq]: GroqIcon, // <-- Add this line

};

**3. Configure the Endpoint on the Backend**

For the frontend to display "Groq" and "Compound," you need to add its configuration to the backend. While I can't modify the files directly, I can show you what you'll need to add to your librechat.yaml file. The server reads this file to populate the endpoint options.

* **File:** librechat.yaml
* **Change:** Add a new entry for groq.

# librechat.yaml

endpoints:

#... other endpoints

groq:

# This will make the settings gear appear. "user\_provided" means the user must enter a key.

apiKey: "user\_provided"

# Defines the models available under Groq

models:

- name: "compound" # The name of your model

# You can add other properties here as needed

context: "16385"