4. OpenRouter Large Model API Aggregation Platform Text Interaction

4. OpenRouter Large Model API Aggregation Platform Text Interaction
    Concept Introduction
        1.1 What is “OpenRouter”?
    2. Practical Operation
        2.1 Obtaining API Credentials
        2.2 Illustrated Workflow
        2.3 Configuration File Modification

Concept Introduction

1.1 What is “OpenRouter”?

OpenRouter is a **large model API aggregation platform**. You can think of it as a "universal remote control for large models". Normally, if you want to use OpenAI's GPT-4, Anthropic's Claude-3, and Google's Gemini, you need to register accounts for three different platforms, obtain three sets of API keys, and write three different sets of calling code for their respective API formats.

OpenRouter solves this problem. It provides a **unified API interface** (compatible with OpenAI's API format). You only need one OpenRouter API key to call all the above models and hundreds of other open-source and closed-source models through this single entry point.

The **core advantage** of integrating OpenRouter into the `largemodel` project is:

- **Extreme flexibility**: You can seamlessly switch between any model such as GPT, Claude, and Llama by modifying a model name in the configuration file, without changing any code.
- **Cost-Effectiveness:** OpenRouter provides clear pricing for each model, allowing you to easily compare and select the most cost-effective model for specific tasks.
- **Explore Cutting-Edge Models:** Many new and excellent open-source models are included in OpenRouter first, allowing developers to quickly try them out.

In the `largemodel` project, you can enable this powerful aggregation platform by configuring the `llm_platform` parameter to `openrouter`.

2. Practical Operation

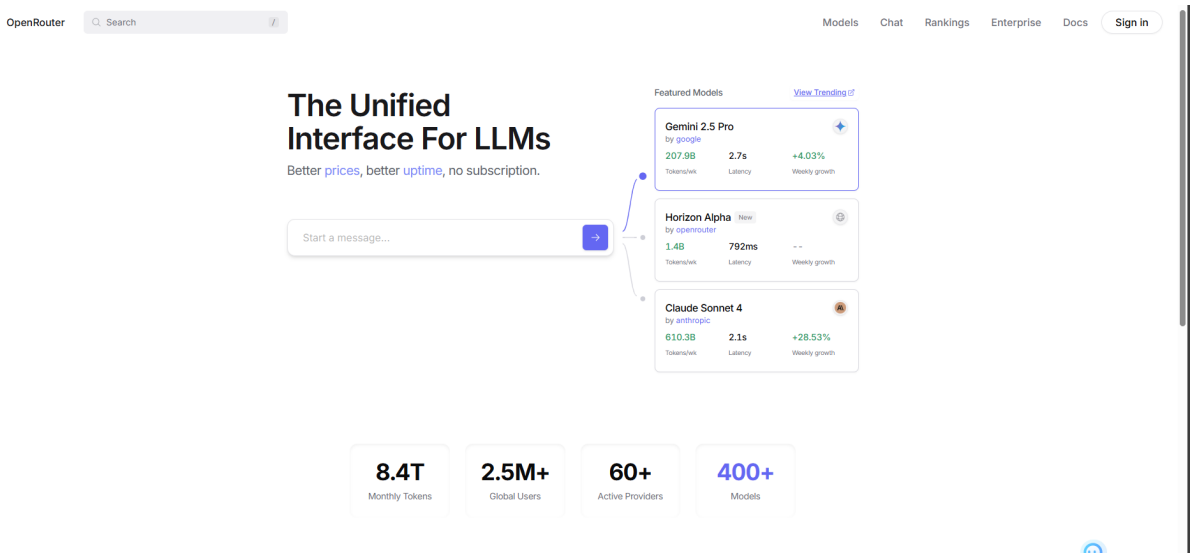
2.1 Obtaining API Credentials

1. **Register for OpenRouter:** Visit OpenRouter.ai, register an account using Google or your email address. If the connection is invalid, you will need to search for the "OpenRouter" platform yourself.
2. **Recharge:** OpenRouter uses a prepaid model. You need to recharge a small amount (e.g., $5) on the "Credits" page to start using it. Alternatively, you can use its free models, but there are rate limits.
3. **Create API Key**: On the "Keys" page, click "Create Key" to generate a new API Key, and **copy and save it immediately**.

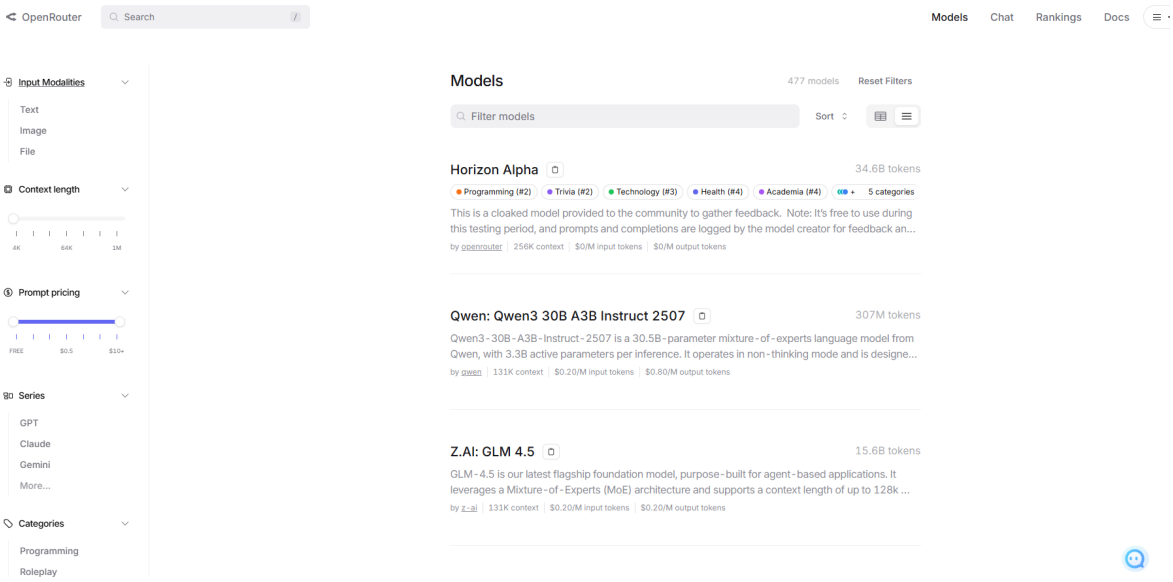
2.2 Illustrated Workflow

1. First, go to the following website. If the connection is invalid, you need to search for the "OpenRouter" platform and then look for keywords related to large models.

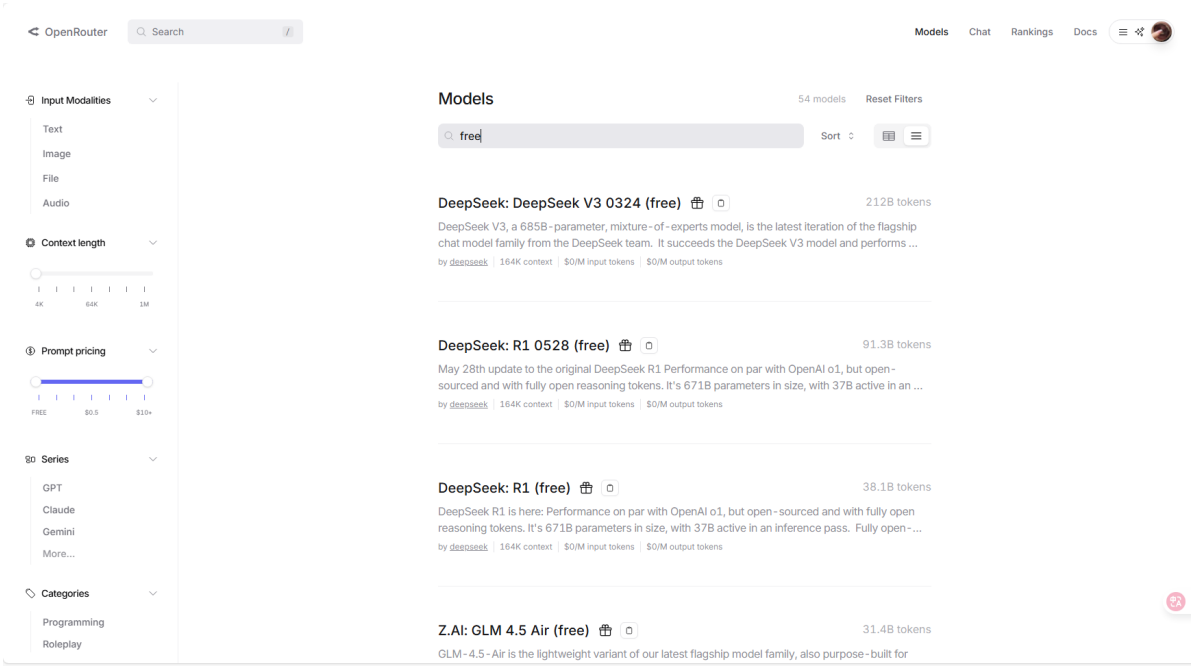
[OpenRouter]



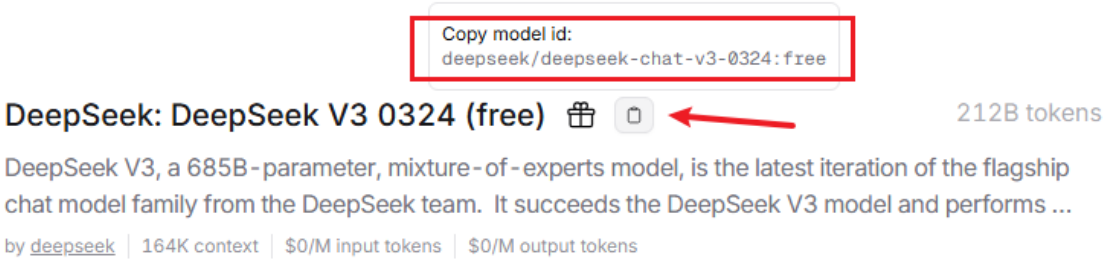
2. Register and log in to an account.
3. After registering an account, click to enter the model marketplace.



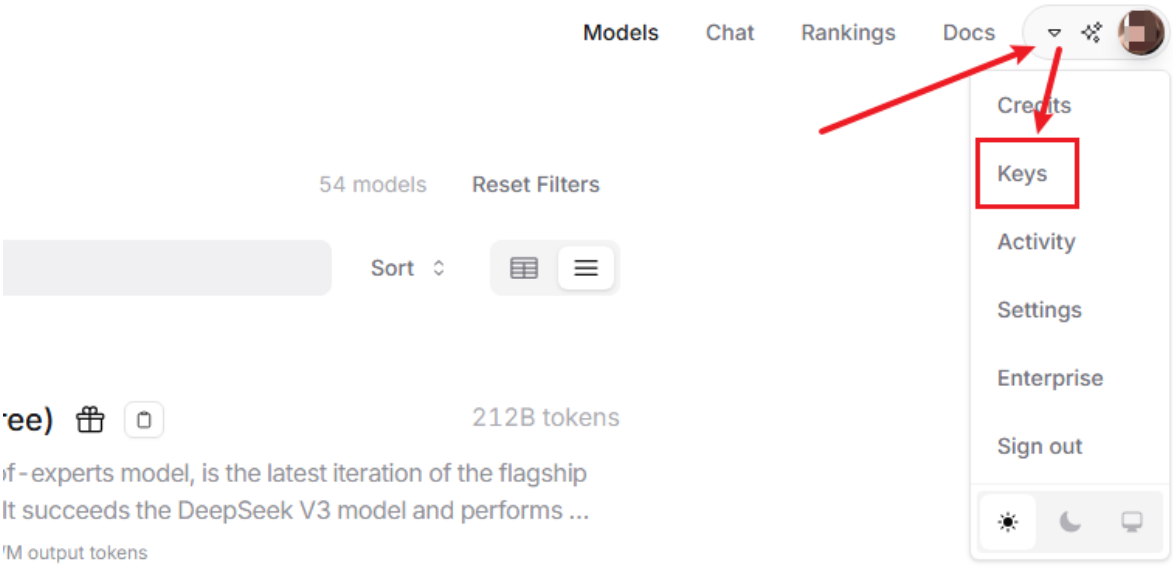
4. You can enter "free" in the search bar to find models that can be used for free.



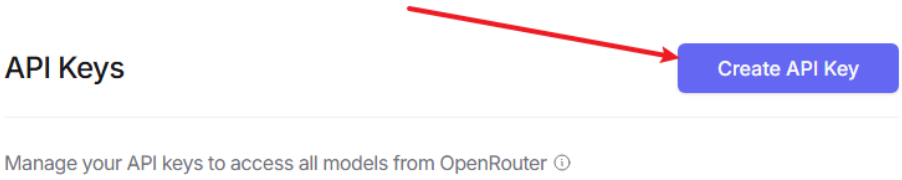
5. Hover your mouse over the clipboard icon next to the model. You'll see the name you'll use to call the model. Click it to copy it.



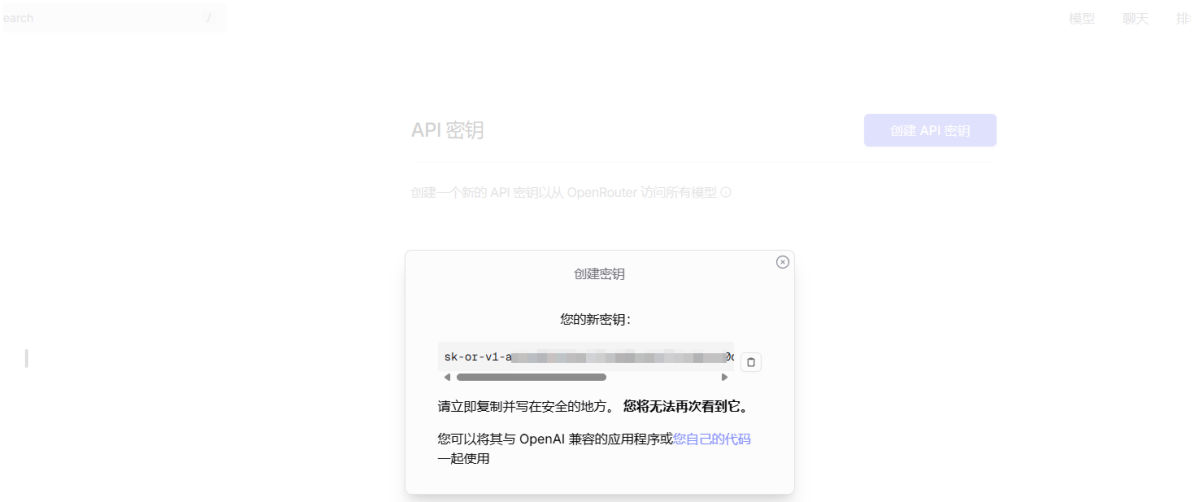
6. Move your mouse to the top right corner next to your avatar. It will expand. Click "Keys" to create a key.



7. Click to apply for an API key



8. After applying, you will receive an API key.



2.3 Configuration File Modification

1. **Open** `large_model_interface.yaml`:

```
vim ~/yahboom_ws/src/largemodel/config/large_model_interface.yaml
```

Then, in openrouter_api_key, change sk-xxxxxxxxxx to the API key you just copied.

openrouter_model is used to configure the large model for dialogue.

```
# OpenRouter Platform Configuration
openrouter_api_key: "sk-xxxxxxxxxxxxxxxxxxxxx"
openrouter_model: "moonshotai/kimi-k2:free" # The model used, such as
"google/gemini-pro-vision"
```

2. **Open** `yahboom.yaml`:

```
vim ~/yahboom_ws/src/largemodel/config/yahboom.yaml
```

3. **Switch Platform**: Change the `llm_platform` parameter to `openrouter`.

```
llm_platform: 'openrouter'
```

After switching the platform parameters, return to the workspace and compile again for the changes to take effect.