## **Vulnerability Name:**

Xinference has an unauthorized access vulnerability

## **Manufacturer:**

Xinference belongs to Hangzhou Future Speed Technology Co., Ltd.

## **Introduction to Xinference**

Xorbits Inference (Xinference) is an open-source platform designed to simplify the operation and integration of various AI models. With Xinference, you can use any open-source LLMs (Large Language Models), embedding models, and multimodal models to perform inference in cloud or local environments, and create powerful AI applications.

GitHub: [https://github.com/xorbitsai/inference](https://github.com/xorbitsai/inference" \t "https://www.doubao.com/chat/_blank)  
Official Documentation: [https://inference.readthedocs.io/zh-cn/latest/index.html](https://inference.readthedocs.io/zh-cn/latest/index.html" \t "https://www.doubao.com/chat/_blank)

## **Version:**

All versions of the Xinference open-source framework (when access authentication is not set)

## **Vulnerability Description:**

Xinference does not set up authentication and access control functions by default. When Xinference starts a local service and sets the host to 0.0.0.0, attackers can remotely invoke Xinference's API interfaces and access the Web page, performing malicious operations including but not limited to stealing sensitive model assets, abusing model computing resources, maliciously deleting models, and causing denial-of-service attacks.

Xinference without authentication and access control functions and exposed to the public Internet is vulnerable to this issue.

## **Vulnerability Code Analysis**

1. ****Local Deployment and Enablement of Xinference****

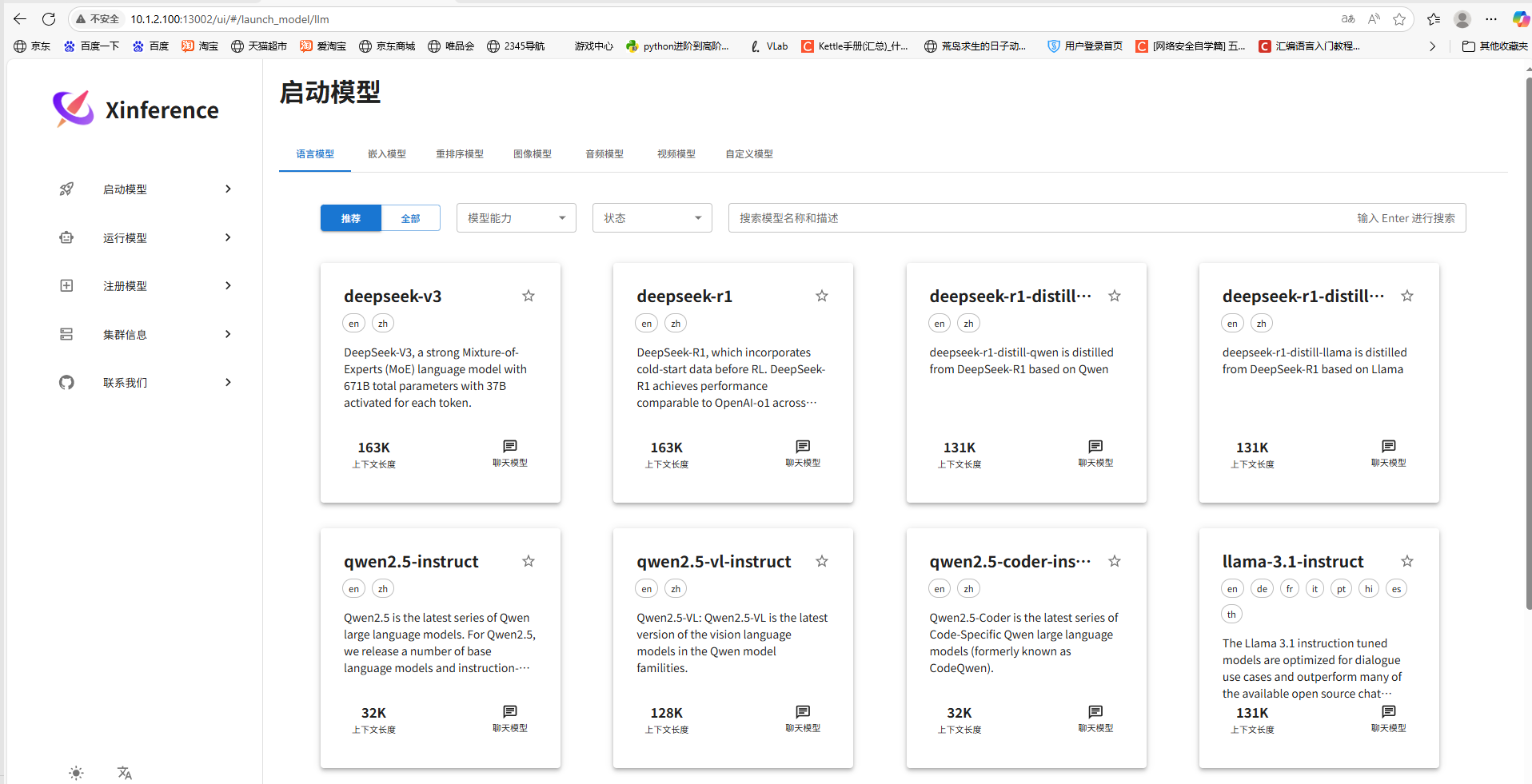
pip install "xinference[all]"

1. ****Command to Run Xinference Locally Under Default Configuration****

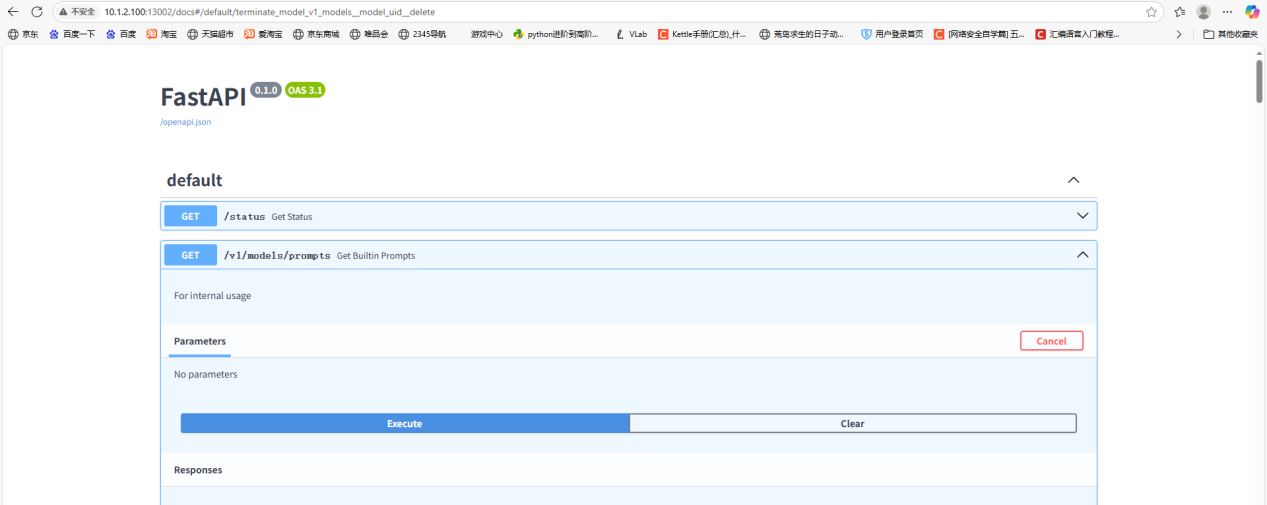
inference-local --host 0.0.0.0 --port 9997

1. For Xinference Exposed to Public Network  
   Any user can access the Xinference web interface and FastAPI services.

Web access route: /ui/#/launch\_model/llm



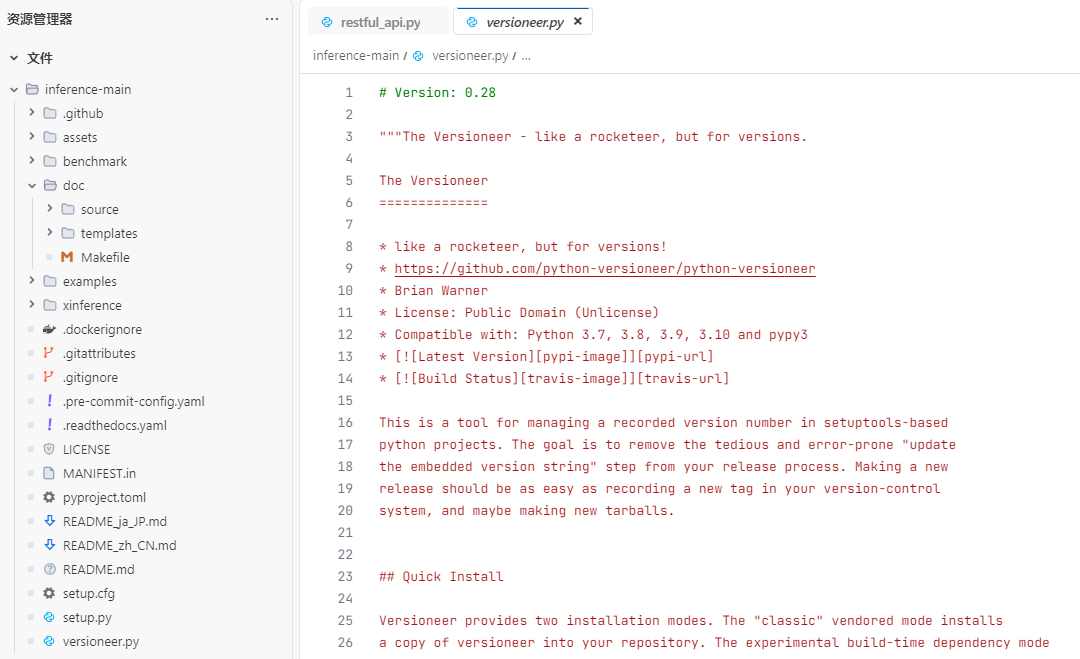
The routing address for accessing FastAPI: /docs



It should be noted that under the default configuration of Xinference, any user can access the web page functions and FastAPI interfaces to pull, view, delete, tamper with models, and perform other operations.

### **Source Code Analysis**

The version of the Xinference source code downloaded in this example is 0.28.



Entering the xinference/api/restful\_api.py source code file, one can observe the relevant routes and interfaces of FastAPI.



By analyzing the source code, it can be seen that: except for routes such as /v1/models/prompts, /v1/models/vllm-supported, and /v1/cluster/info, other routes all have identity dependencies for permission verification.



Analyzing the identity dependencies of route permission verification again, the source code is as follows.

Here is the interpretation of the source code: To execute the /v1/models/{model\_uid} route, it is necessary to pass through dependencies. In dependencies:

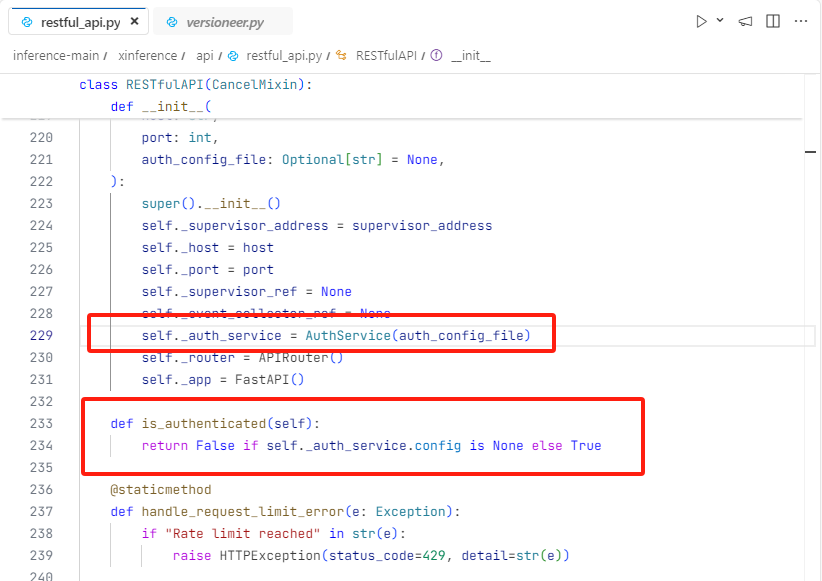
* If the value of self.is\_authenticated() is True (i.e., when system authentication is enabled), a permission check dependency is added to the API route, requiring that users accessing the route must have the models:stop permission.
* If system authentication is not enabled, no permission check is performed.



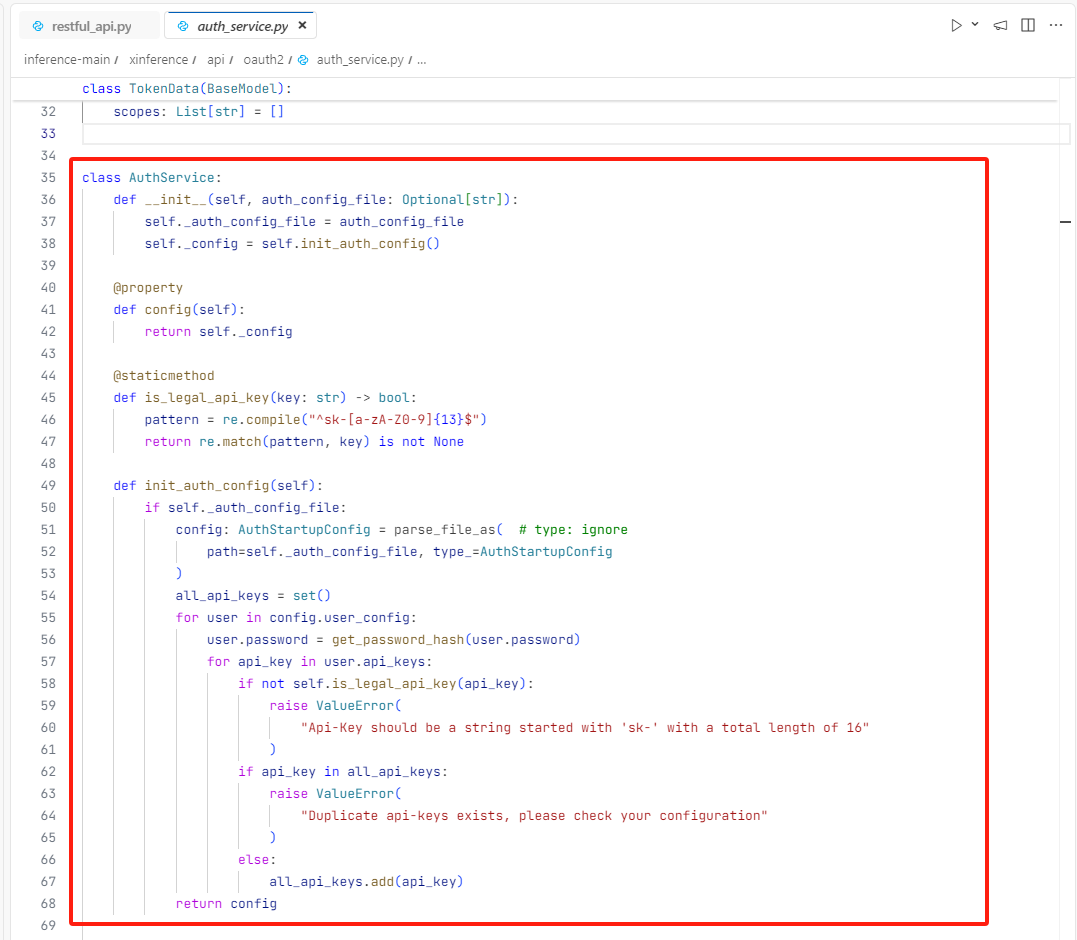
Continuing to trace the self.is\_authenticated() method, the code is as follows:

It is found that the value of self.is\_authenticated() is related to self.\_auth\_service.config. If self.\_auth\_service.config is empty, it returns False.

Through self.\_auth\_service, the AuthService(auth\_config\_file) method can be traced.



The AuthService method is located in xinference/api/oauth2, and the relevant source code is as follows:



Through the above source code analysis, we can further draw the following conclusions:

* The value of \_auth\_service.config is the value of self.\_config in the AuthService class, and the value of self.\_config is returned by self.init\_auth\_config().
* The value returned by self.init\_auth\_config() is determined by auth\_config\_file.

****Key point****: If auth\_config\_file is empty, the value returned by self.init\_auth\_config() is also empty.

What is the value of auth\_config\_file in the RESTful API route? Further inspection shows that it is empty by default.



Therefore, the following conclusions can be drawn:

By default, the auth\_config\_file in Xinference is empty, resulting in the default absence of permission settings.

Under the default configuration of Xinference, attackers can access any FastAPI route without authorization, enabling operations such as model invocation and model deletion.

## Vulnerability Reproduction

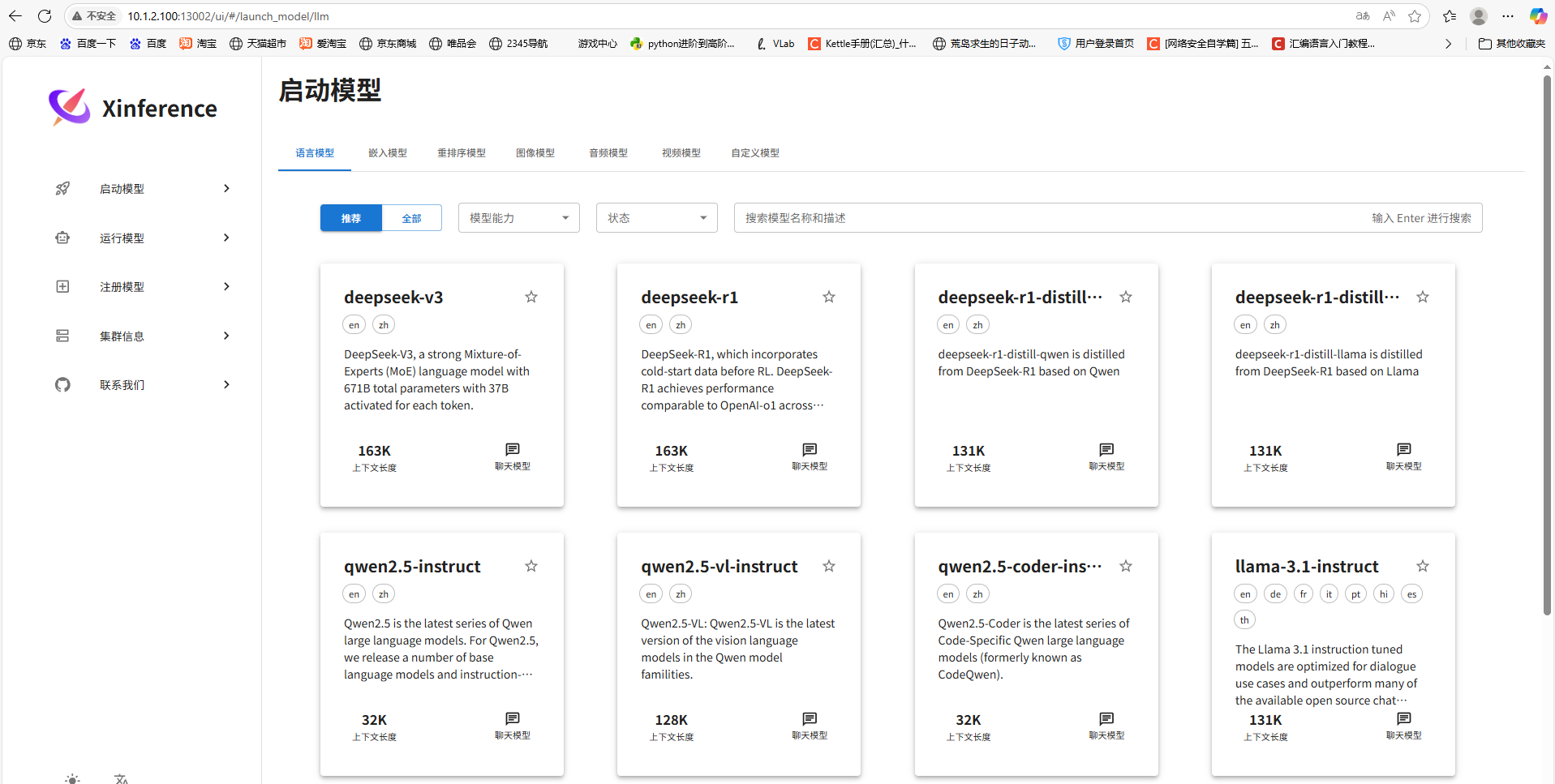
1. ****Local Deployment and Enablement of Xinference****

pip install "xinference[all]"

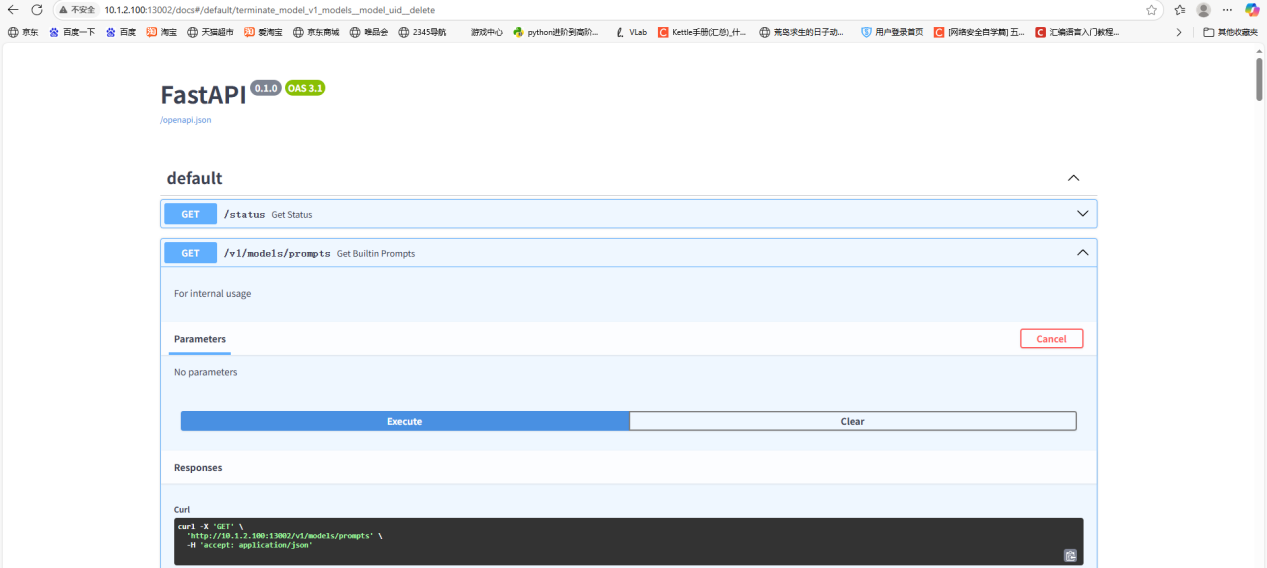
1. ****Command to Run Xinference Locally Under Default Configuration****

inference-local --host 0.0.0.0 --port 9997

Current IP: 10.1.2.100

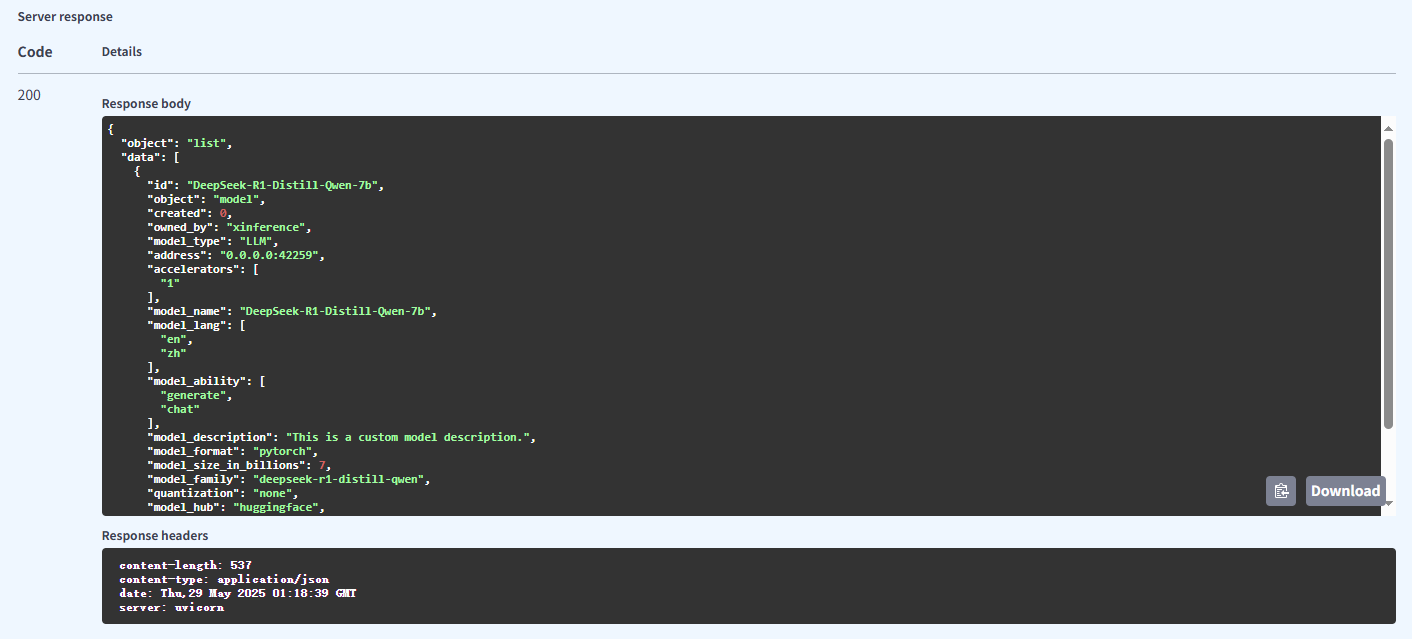


1. Visit [http://10.1.2.100:13002/docs](http://10.1.2.100:13002/docs" \t "https://www.doubao.com/chat/_blank)#, or use API interface calling tools to implement the following steps.



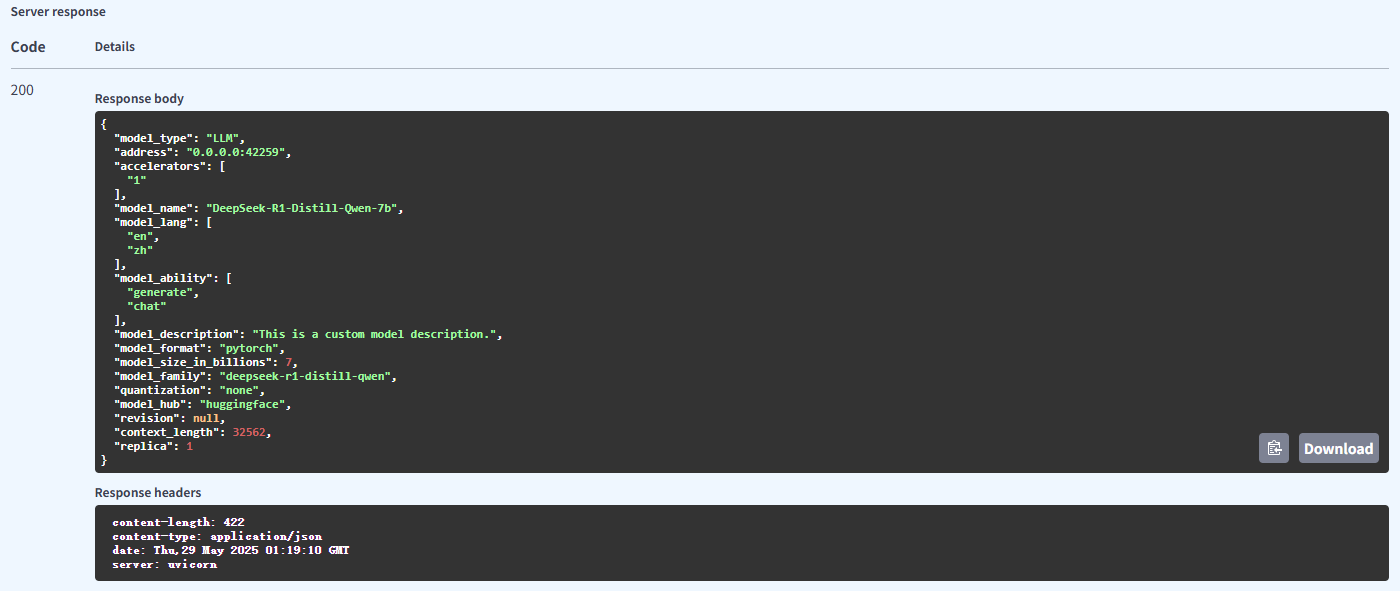
1. View the current model list



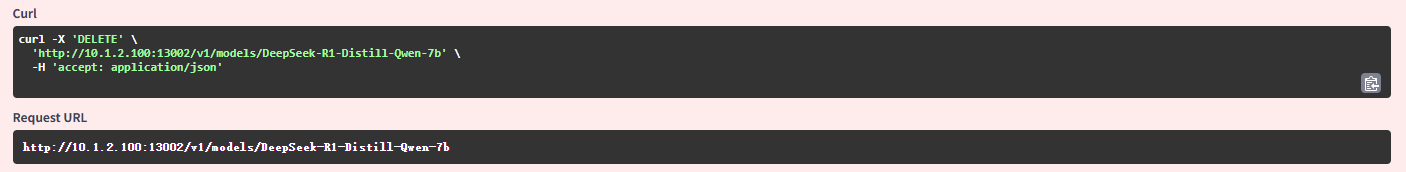


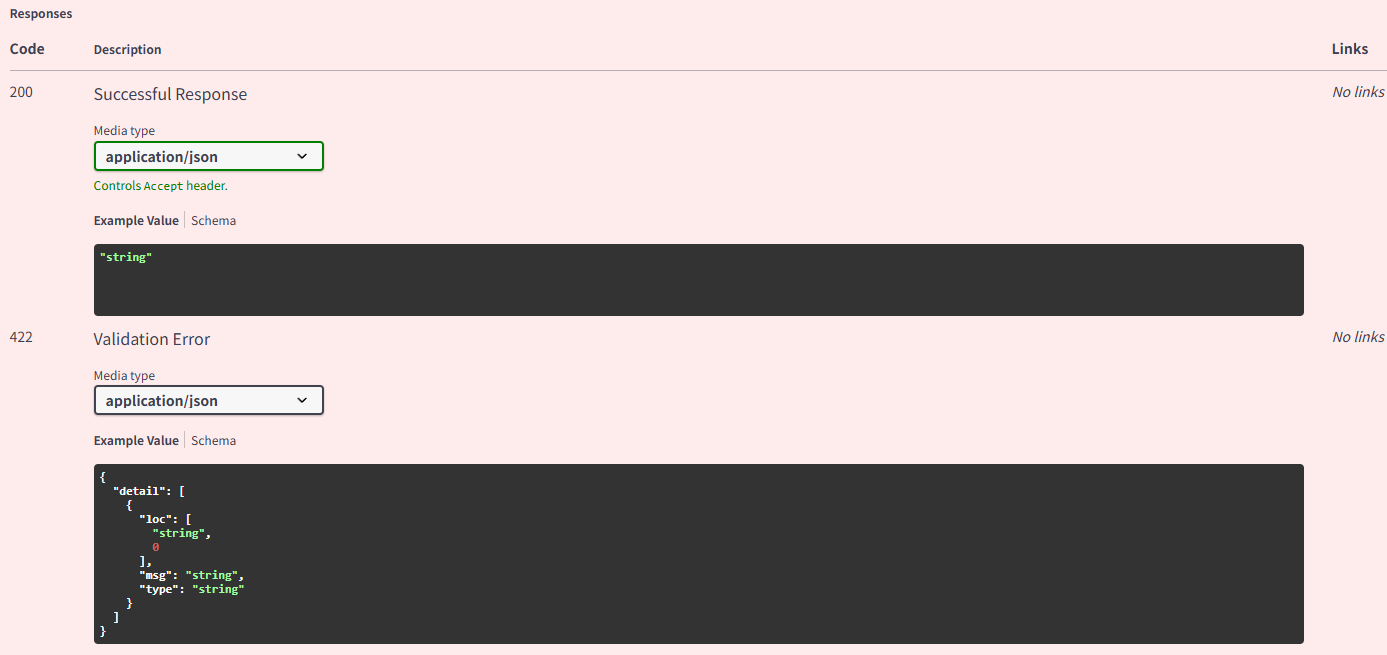
1. Access model-related descriptions





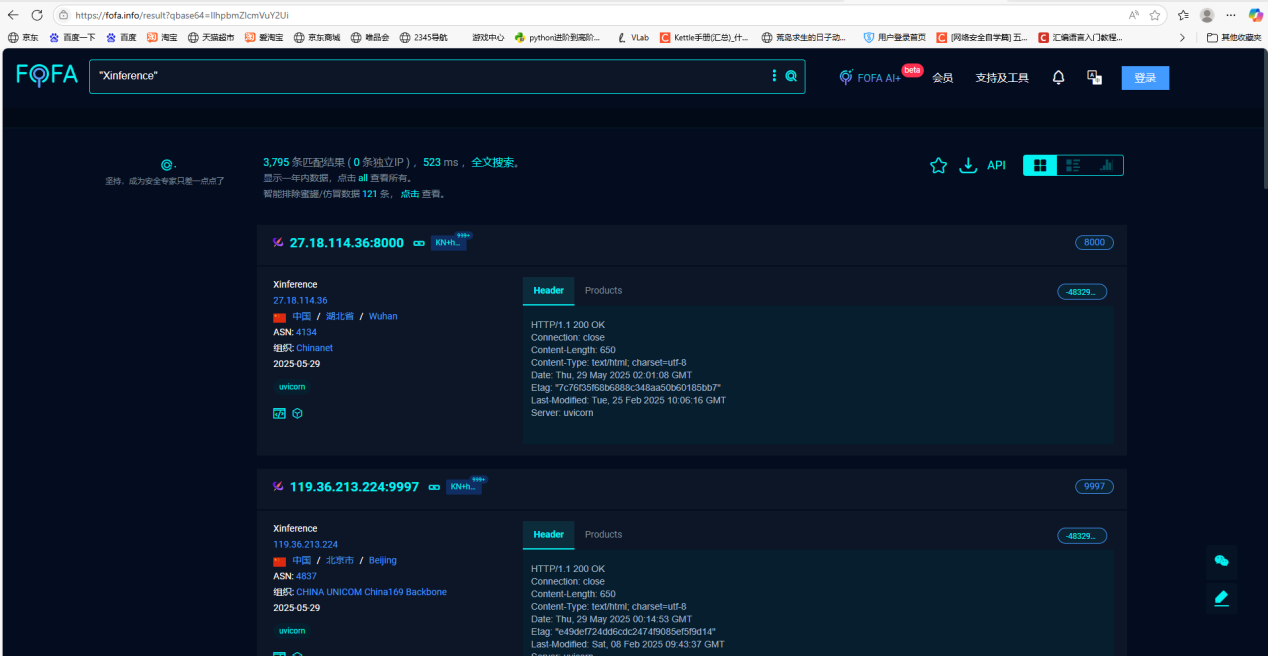
1. Delete the specified model





Further verification of the model interface confirms that the model has indeed been deleted.

## **Asset Collection** Through the "Xinference" command, it was found that there are no fewer than 3,795 assets exposed to the public network, and most of them are accessible.



## **Vulnerability Fix Suggestions**

1. It is necessary to avoid exposing Xinference to the public network.
2. Modify the value of the default configuration auth\_config\_file to prevent it from being empty; or set other permission control methods.