API Documentation for Auth and Log Usage

This documentation outlines how to interact with the authentication and log usage services using Python's `requests` library.

Requirements

Before you start, ensure Python and the `requests` library are installed:

```
pip install requests
```

Authentication API

Endpoint Description

This endpoint authenticates users or services.

HTTP Method: POST

URL: `/api/guard/auth`

Headers

- **Content-Type**: `application/json` - Indicates the media type of the resource.

- **Authorization**: `sk-xxx` - API key for access control.

- **SecretKey**: `xxx` - A client-specific secret key for additional security.

- **Swarms-Organization**: _(Optional)_ - Header provided by the user for organization-specific configuration.

Request Payload

```json
{
  "model": "cogvlm-chat-17b",
  "messages": [
    {
      "role": "system",
      "content": "You are a poetic assistant, skilled in explaining complex programming concepts with creative flair."
    },
    {
      "role": "user",
      "content": "Compose a poem that explains the concept of recursion in programming."
    }
  ]
}
```

- **model**: Model identifier used for the API.

- **messages**: An array containing message objects.

  - **role**: Role of the message sender (`system` or `user`).
  - **content**: Text content of the message.

Python Example

```python
import requests


def authenticate():
    url = "http://localhost:3000/api/guard/auth"
    headers = {
        "Content-Type": "application/json",
        "Authorization": "sk-xxx",
        "SecretKey": "xxx"
    }
    payload = {
        "model": "cogvlm-chat-17b",
        "messages": [
            {"role": "system", "content": "You are a poetic assistant, skilled in explaining complex programming concepts with creative flair."},
            {"role": "user", "content": "Compose a poem that explains the concept of recursion in programming."}
        ]
    }
    response = requests.post(url, json=payload, headers=headers)
    return response.json()
```

~~~

Log Usage API

Endpoint Description

This endpoint logs usage statistics of the API.

HTTP Method: POST

URL: `/api/guard/log-usage`

Headers

- **Content-Type**: `application/json`
- **Authorization**: `sk-xxx`
- **SecretKey**: `xxx`

Request Payload

```json
{
  "model": "cogvlm-chat-17b",
  "temperature": 0.7,
  "top_p": 0.9,
  "input_cost": 0.5,
~~~

```
  "output_cost": 0.3,
  "total_cost": 0.8,
  "input_tokens": 100,
  "output_tokens": 80,
  "max_tokens": 200,
  "messages": [
    {
      "role": "system",
       "content": "You are a poetic assistant, skilled in explaining complex programming concepts with creative flair."
    },
    {
      "role": "user",
      "content": "Compose a poem that explains the concept of recursion in programming."
    }
  ]
}
```

- **temperature**: Controls randomness in output generation.
- **top_p**: Nucleus sampling cutoff.
- **input_cost**: Cost per input token.
- **output_cost**: Cost per output token.
- **total_cost**: Total cost calculated from inputs and outputs.
- **input_tokens**: Number of input tokens.
- **output_tokens**: Number of output tokens.

- **max_tokens**: Maximum tokens allowed in the output.

Python Example

```python
import requests

def log_usage():
    url = "http://localhost:3000/api/guard/log-usage"
    headers = {
        "Content-Type": "application/json",
        "Authorization": "sk-xxx",
        "SecretKey": "xxx"
    }
    payload = {
        "model": "cogvlm-chat-17b",
        "temperature": 0.7,
        "top_p": 0.9,
        "input_cost": 0.5,
        "output_cost": 0.3,
        "total_cost": 0.8,
        "input_tokens": 100,
        "output_tokens": 80,
        "max_tokens": 200,
        "messages": [
            {"role": "system", "content": "You are a poetic assistant, skilled in explaining complex
programming concepts with creative flair."},
                {"role": "user", "content": "Compose a poem that explains the concept of recursion in
programming."}
        ]
    }
    response = requests.post(url, json=payload, headers=headers)
    return response.json()
```