```python
import os

from pydantic import BaseModel

from typing import Optional, Dict

from uuid import UUID


# from swarms_cloud.auth_with_swarms_cloud import supabase_client_init


class ModelAPILogEntry(BaseModel):
    id: Optional[UUID] = None

    created_at: Optional[str] = None

    user_id: Optional[UUID] = None

    api_key_id: Optional[UUID] = None

    model_id: Optional[UUID] = None

    input_tokens: Optional[int] = None

    output_tokens: Optional[int] = None

    all_cost: Optional[float] = None

    input_cost: Optional[float] = None

    output_cost: Optional[float] = None

    messages: Optional[Dict] = None

    status: Optional[int] = None

    temperature: Optional[float] = None

    top_p: Optional[float] = None

    echo: Optional[bool] = None

    max_tokens: Optional[int] = None
```

```python
    stream: Optional[bool] = None

    repetition_penalty: Optional[float] = None


async def log_to_supabase(
    table_name: str = "swarms_cloud_api_key_activities",

    entry: ModelAPILogEntry = None,

    # supabase: Client = supabase_client_init,

):
    """

    Logs an entry to Supabase.


    Args:

        table_name (str, optional): The name of the table to insert the entry into. Defaults to
"swarms_cloud_api_key_activities".

        entry (LogEntry, optional): The log entry to be inserted. Defaults to None.

        supabase (Client, optional): The Supabase client instance. Defaults to None.


    Returns:

        dict: The response from the Supabase insert operation, or an error message if an exception
occurs.
    """

    from supabase import create_client


    # Supabase client

    supabase = create_client(
```

```python
        supabase_url=os.getenv("SUPABASE_URL"),

        supabase_key=os.getenv("SUPABASE_KEY"),

    )

    try:

        response = supabase.table(table_name).insert(entry.model_dump()).execute()

        return response

    except Exception as error:

        print(f"Error logging to Supabase: {error}")

        return error
```