```
from __future__ import annotations


import time
import uuid
from typing import List, Optional
from typing import Any
from pydantic import BaseModel, Field
from swarms.schemas.base_schemas import (
    AgentChatCompletionResponse,
)
from typing import Union



def get_current_time():
    return time.strftime("%Y-%m-%d %H:%M:%S")



uuid_hex = uuid.uuid4().hex



class Step(BaseModel):
    step_id: Optional[str] = Field(
        default_factory=lambda: uuid.uuid4().hex,
        description="The ID of the task step.",
        examples=["6bb1801a-fd80-45e8-899a-4dd723cc602e"],
    )
    time: Optional[float] = Field(
        default_factory=get_current_time,
        description="The time taken to complete the task step.",
    )
    response: Optional[AgentChatCompletionResponse]


class ManySteps(BaseModel):
    agent_id: Optional[str] = Field(
        ...,
        description="The ID of the agent.",
        examples=["financial-agent-1"],
    )
    agent_name: Optional[str] = Field(
        ...,
        description="The ID of the agent.",
        examples=["financial-agent-1"],
    )
    task: Optional[str] = Field(
        ...,
        description="The name of the task.",
        examples=["Write to file"],
    )
    max_loops: Optional[Any] = Field(
        ...,
        description="The number of steps in the task.",
        examples=[3],
    )
    run_id: Optional[str] = Field(
        uuid.uuid4().hex,
        description="The ID of the task this step belongs to.",
        examples=["50da533e-3904-4401-8a07-c49adf88b5eb"],
    )
    steps: Optional[List[Union[Step, Any]]] = Field(
        [],
        description="The steps of the task.",
    )
    full_history: Optional[str] = Field(
        ...,
        description="The full history of the task.",
        examples=[
            "I am going to use the write_to_file command and write"
            " Washington to a file called output.txt"
            " <write_to_file('output.txt', 'Washington')"
        ],
    )
    total_tokens: Optional[int] = Field(
        ...,
        description="The total number of tokens generated.",
        examples=[7894],
    )
    stopping_token: Optional[str] = Field(
        ...,
        description="The token at which the task stopped.",
    )
    interactive: Optional[bool] = Field(
        ...,
        description="The interactive status of the task.",
        examples=[True],
    )
    dynamic_temperature_enabled: Optional[bool] = Field(
        ...,
        description="The dynamic temperature status of the task.",
        examples=[True],
    )
```