

```
from pydantic import BaseModel, Field

from transformers import AutoModelForCausalLM, AutoTokenizer


from swarms import ToolAgent

from swarms.tools.json_utils import base_model_to_json


# Model name

model_name = "ai21labs/Jamba-v0.1"


# Load the pre-trained model and tokenizer

model = AutoModelForCausalLM.from_pretrained(

    model_name,

    device_map="auto",

)


# Load the pre-trained model and tokenizer

tokenizer = AutoTokenizer.from_pretrained(model_name)


# Initialize the schema for the person's information

class APIExampleRequestSchema(BaseModel):

    endpoint: str = Field(

        ..., description="The API endpoint for the example request"

    )

    method: str = Field(

        ..., description="The HTTP method for the example request"
```

```

)

headers: dict = Field(
    ..., description="The headers for the example request"
)

body: dict = Field(
    ..., description="The body of the example request"
)

response: dict = Field(
    ...,
    description="The expected response of the example request",
)

```

```

# Convert the schema to a JSON string

```

```

api_example_schema = base_model_to_json(APIExampleRequestSchema)

```

```

# Convert the schema to a JSON string

```

```

# Define the task to generate a person's information

```

```

task = "Generate an example API request using this code:\n"

```

```

# Create an instance of the ToolAgent class

```

```

agent = ToolAgent(
    name="Command R Tool Agent",
    description=(
        "An agent that generates an API request using the Command R"
        " model."
    )
)

```

```
),  
  
model=model,  
  
tokenizer=tokenizer,  
  
json_schema=api_example_schema,  
  
)  
  
# Run the agent to generate the person's information  
  
generated_data = agent(task)  
  
# Print the generated data  
  
print(f"Generated data: {generated_data}")
```