```
import os

from ai21 import AI21Client
from ai21.models.chat import ChatMessage
from dotenv import load_dotenv

from swarms import BaseLLM

load_dotenv()



class Jamba(BaseLLM):
    def __init__(
        self,
        api_key: str = os.getenv("AI21_API_KEY"),
        temperature: int = 0.8,
        max_tokens: int = 200,
    ):
        """
        Initializes the Jamba class with the provided API key.

        Args:
            api_key (str): The API key for the AI21Client.
        """
        os.environ["AI21_API_KEY"] = api_key
        self.api_key = api_key
```

```
    self.temperature = temperature
    self.max_tokens = max_tokens
    self.client = AI21Client()

def run(self, prompt: str, *args, **kwargs) -> str:
    """
    Generates a response for the given prompt using the AI21 model.

    Args:
        prompt (str): The prompt for generating the response.

    Returns:
        str: The generated response.

    Raises:
        Exception: If there is an issue with the API request.
    """
    try:
        response = self.client.chat.completions.create(
            model="jamba-instruct-preview",  # Latest model
            messages=[ChatMessage(role="user", content=prompt)],
            temperature=self.temperature,
            max_tokens=self.max_tokens,
            *args,
            **kwargs,
        )
```

```
        return response.choices[0].message.content
    except Exception as e:
        print(f"Error: {e}")
        raise
```