```
import tiktoken
from fastapi import HTTPException
import asyncio

# logger.info("Starting the agent API server...")



def count_tokens(text: str) -> int:
    """
    Counts the number of tokens in the given text.

    Args:
        text (str): The input text to count tokens from.

    Returns:
        int: The number of tokens in the text.

    Raises:
        HTTPException: If there is an error counting tokens.
    """
    try:
        # Get the encoding for the specific model
        enc = tiktoken.get_encoding("cl100k_base")

        # Encode the text
        tokens = enc.encode(text)
```

```
        # Count the tokens
        return len(tokens)
    except Exception as e:
        raise HTTPException(status_code=400, detail=f"Error counting tokens: {e}")


async def count_tokens_async(text: str) -> int:
    """
    Counts the number of tokens in the given text.

    Args:
        text (str): The input text.

    Returns:
        int: The number of tokens in the text.

    Raises:
        HTTPException: If there is an error counting tokens.
    """
    loop = asyncio.get_event_loop()
    try:
        # Get the encoding for the specific model
        enc = await loop.run_in_executor(None, tiktoken.get_encoding, "cl100k_base")

        # Encode the text
```

```
        tokens = await loop.run_in_executor(None, enc.encode, text)

        # Count the tokens
        return len(tokens)
    except Exception as e:
        raise HTTPException(status_code=400, detail=f"Error counting tokens: {e}")
```