

Documentação do arquivo openAIclass.cls

Introdução

O arquivo `openAIclass.cls` é uma classe Apex que permite a integração com a API OpenAI para realizar operações de conclusão de texto. Ele utiliza o modelo `text-davinci-003` para gerar respostas baseadas em um prompt fornecido.

Descrição

Esta classe contém um método estático chamado `textcompletionCeck`, que envia uma solicitação HTTP para a API OpenAI e retorna a resposta gerada pelo modelo. O método é marcado com a anotação `@AuraEnabled`, permitindo que ele seja chamado a partir de componentes Lightning ou outras interfaces que suportem chamadas de métodos Apex.

Estrutura

A estrutura do arquivo consiste em:

- Uma classe pública chamada `openAIclass`.
- Um método estático chamado `textcompletionCeck`.
- Configuração de cabeçalhos HTTP e corpo da solicitação para interagir com a API OpenAI.
- Manipulação da resposta da API para extrair o texto gerado.

Dependências

- A classe depende da API OpenAI para realizar operações de conclusão de texto.
- Utiliza a classe `Http` e `HttpRequest` do Apex para enviar solicitações HTTP.
- Depende de uma classe auxiliar chamada `openaiV0` para processar a resposta JSON da API.

Imports

Não há importações explícitas no Apex, mas a classe utiliza as seguintes classes nativas:

- `Http`
- `HttpRequest`
- `HttpResponse`

Variáveis

Variáveis Locais

- `Http http`: Instância da classe `Http` usada para enviar solicitações HTTP.
- `HttpRequest request`: Instância da classe `HttpRequest` usada para configurar a solicitação HTTP.
- `string requestBody`: Corpo da solicitação em formato JSON, contendo os parâmetros necessários para a API OpenAI.
- `HttpResponse response`: Instância da classe `HttpResponse` usada para armazenar a resposta da API.
- `openaiV0 responseRetrived`: Instância de uma classe auxiliar para processar a resposta JSON.

Métodos

`textcompletionCeck`

Descrição

Este método realiza uma chamada à API OpenAI para gerar uma conclusão de texto com base no prompt fornecido.

Parâmetros

- `string texttoCheck`: O texto de entrada que será enviado como prompt para a API OpenAI.

Retorno

- Retorna uma `string` contendo o texto gerado pela API OpenAI.

Funcionamento

1. Cria uma instância de `Http` e `HttpRequest`.
2. Configura o endpoint da API OpenAI e define o método HTTP como `POST`.
3. Adiciona os cabeçalhos necessários, incluindo a autorização e o tipo de conteúdo.
4. Define o corpo da solicitação em formato JSON, especificando o modelo, o prompt, o número máximo de tokens e a temperatura.
5. Envia a solicitação e recebe a resposta.
6. Processa a resposta JSON usando a classe `openaiV0` e retorna o texto gerado.

Exemplo

Aqui está um exemplo de como usar o método `textcompletionCeck`:

```
string prompt = 'Escreva uma introdução sobre inteligência artificial.';
string result = openAIclass.textcompletionCeck(prompt);
System.debug('Texto gerado pela API OpenAI: ' + result);
```

Diagrama de Dependência

O diagrama abaixo ilustra as dependências entre as classes e métodos:

```
classDiagram
    class openAIclass {
        +textcompletionCeck(string texttoCheck) : string
    }
    class Http {
        +send(HttpRequest request) : HttpResponse
    }
    class HttpRequest {
        +setEndpoint(string url)
        +setMethod(string method)
        +setHeader(string key, string value)
        +setBody(string body)
    }
    class HttpResponse {
        +getBody() : string
    }
    class openaiV0 {
        +parse(string json) : openaiV0
        +choices : List<Choice>
    }
    class Choice {
        +text : string
    }
    openAIclass --> Http
    openAIclass --> HttpRequest
    openAIclass --> HttpResponse
    openAIclass --> openaiV0
    openaiV0 --> Choice
```

Notas

- Certifique-se de substituir a chave de API (`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`) e a organização (`org-HRp36KGU9iP2IeUUzw0MyQvi`) por valores válidos e seguros antes de usar em produção.

- A classe `openaiV0` não está definida neste arquivo. Certifique-se de que ela esteja implementada corretamente para processar a resposta JSON da API.

Vulnerabilidades

- **Exposição de Credenciais:** A chave de API e o identificador da organização estão expostos diretamente no código. Isso pode levar a comprometimento de segurança. Recomenda-se armazenar essas informações em um local seguro, como o Gerenciador de Configuração do Salesforce.
- **Limitação de Tokens:** O número máximo de tokens é definido como 4000. Certifique-se de que isso esteja dentro dos limites da API OpenAI para evitar erros.
- **Manipulação de Erros:** O código não trata erros que podem ocorrer durante a solicitação HTTP ou o processamento da resposta. Adicione tratamento de exceções para melhorar a robustez.