

# Documentação do arquivo openAIclass.cls

## Introdução

O arquivo `openAIclass.cls` é uma classe Apex que permite a integração com a API da 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 da OpenAI e retorna o texto gerado pela IA. O método utiliza a biblioteca HTTP nativa do Apex para realizar a comunicação com a API.

## Estrutura

A classe possui uma estrutura simples, com um único método estático que realiza a funcionalidade principal de comunicação com a API da OpenAI.

## Dependências

- API da OpenAI (<https://api.openai.com/v1/completions>)
- Classe `openaiV0` para manipulação da resposta JSON retornada pela API.

## Imports

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

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

## Variáveis

### Variáveis Locais

- `Http http`: Instância da classe `Http` para enviar solicitações HTTP.
- `HttpRequest request`: Instância da classe `HttpRequest` para configurar a solicitação HTTP.
- `string requestBody`: Corpo da solicitação em formato JSON, contendo os parâmetros para a API da OpenAI.
- `HttpResponse response`: Instância da classe `HttpResponse` para armazenar a resposta da API.
- `openaiV0 responseRetrived`: Objeto para manipular e acessar os dados retornados pela API.

## Métodos

### `textcompletionCeck`

#### Descrição

Este método realiza uma solicitação HTTP POST para a API da OpenAI, enviando um prompt de texto e retornando a resposta gerada pelo modelo de IA.

#### Parâmetros

- `string textttoCheck`: Texto que será enviado como prompt para a API da OpenAI.

## Retorno

- string: Texto gerado pela API da OpenAI com base no prompt fornecido.

## Funcionamento

1. Cria uma instância de Http e HttpRequest.
2. Configura o endpoint da API, método HTTP, cabeçalhos e corpo da solicitação.
3. Envia a solicitação HTTP e recebe a resposta.
4. Analisa o corpo da resposta JSON usando a classe openaiV0.
5. Retorna o texto gerado pela API.

## Exemplo

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

## Diagrama de Dependência

```
classDiagram
    class openAIclass {
        +textcompletionCeck(string texttoCheck) : string
    }
    class Http {
        +send(HttpRequest request) : HttpResponse
    }
    class HttpRequest {
        +setEndpoint(string endpoint)
        +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

- A chave de API (Bearer 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) e a organização (org-HRp36KGU9iP2IeUUzw0MyQvi) estão expostas diretamente no código. Isso pode representar um risco de segurança e deve ser tratado com cuidado.
- O método utiliza o modelo text-davinci-003, que é um dos modelos mais avançados da OpenAI para geração de texto.

## Vulnerabilidades

1. **Exposição de Credenciais:** A chave de API e a organização estão diretamente no código. Isso pode levar ao comprometimento da conta da OpenAI caso o código seja acessado por terceiros.
  - **Solução:** Armazenar as credenciais em um local seguro, como o Custom Settings ou Protected Custom Metadata no Salesforce.

2. **Limitação de Tokens:** O parâmetro `max_tokens` está definido como 4000. Dependendo do prompt, isso pode gerar respostas muito longas e consumir mais recursos da API.
    - **Solução:** Ajustar o valor de `max_tokens` conforme necessário para evitar consumo excessivo.
  3. **Manipulação de Erros:** O código não trata erros que podem ocorrer durante a solicitação HTTP ou na análise da resposta JSON.
    - **Solução:** Implementar tratamento de exceções para garantir que o sistema lide adequadamente com falhas na API ou problemas de rede.
- 

Esta documentação detalha o funcionamento da classe `openAIclass.cls` e fornece informações importantes para sua utilização e manutenção.