

# Documentação do arquivo

## CaseMessageFromCommentController.cls

### Introdução

O arquivo `CaseMessageFromCommentController.cls` contém uma classe Apex que é utilizada para enviar o último comentário público de um caso para um número de telefone associado ao caso. Ele é projetado para ser usado em conjunto com o framework Lightning Aura.

### Descrição

A classe `CaseMessageFromCommentController` possui um método estático chamado `sendLastPublicComment`, que realiza as seguintes ações:

1. Busca o último comentário público associado a um caso específico.
2. Verifica se o caso possui um número de telefone fornecido.
3. Utiliza um serviço externo chamado `SendMessageAtendechat` para enviar o comentário público para o número de telefone.

Este código é útil em cenários onde é necessário comunicar informações de um caso diretamente para um cliente ou usuário final via mensagem.

### Estrutura

A classe possui:

- Um modificador de acesso `public with sharing`, garantindo que as regras de compartilhamento do Salesforce sejam respeitadas.
- Um método estático `sendLastPublicComment` marcado com a anotação `@AuraEnabled`, permitindo que ele seja chamado a partir de componentes Lightning.

### Dependências

A classe depende das seguintes entidades:

- Objeto padrão `CaseComment` para buscar comentários associados a casos.
- Objeto padrão `Case` para buscar o número de telefone associado ao caso.
- Classe externa `SendMessageAtendechat` para enviar mensagens.

### Imports

Não há importações explícitas no código Apex, pois ele utiliza objetos padrão do Salesforce e uma classe externa.

### Variáveis

O código utiliza as seguintes variáveis:

- `comments`: Lista de comentários públicos associados ao caso.
- `c`: Instância do objeto `Case` que contém o número de telefone do caso.

### Métodos

**Método:** `sendLastPublicComment`

**Descrição:**  
Este método é responsável por enviar o último comentário público de um caso para o número de telefone associado ao caso.

**Parâmetros:**

- Id caseId: O ID do caso para o qual o comentário será enviado.

**Funcionamento:**

1. Realiza uma consulta SOQL para buscar o último comentário público do caso.
2. Verifica se há comentários públicos disponíveis. Caso contrário, lança uma exceção AuraHandledException com a mensagem "No public comment found."
3. Realiza outra consulta SOQL para buscar o número de telefone associado ao caso.
4. Verifica se o número de telefone está presente. Caso contrário, lança uma exceção AuraHandledException com a mensagem "Phone number is missing."
5. Chama o método send da classe SendMessageAtendechat para enviar o comentário público para o número de telefone.

**Exceções:**

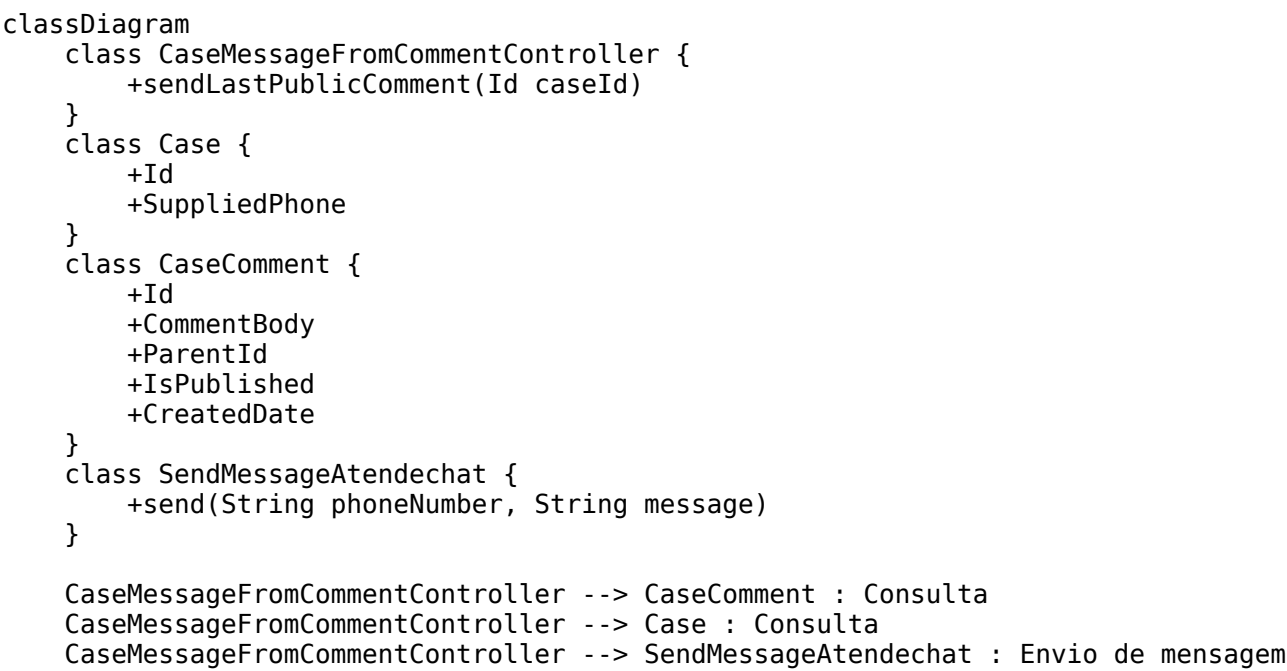
- AuraHandledException: Lançada quando não há comentários públicos ou quando o número de telefone está ausente.

# Exemplo

Aqui está um exemplo de como usar o método sendLastPublicComment:

```
Id caseId = '500XXXXXXXXXXXXXXXXX'; // Substitua pelo ID de um caso válido
try {
    CaseMessageFromCommentController.sendLastPublicComment(caseId);
    System.debug('Comentário enviado com sucesso!');
} catch (AuraHandledException e) {
    System.debug('Erro: ' + e.getMessage());
}
```

# Diagrama de Dependência



# Notas

- Certifique-se de que a classe SendMessageAtendechat esteja implementada corretamente e disponível no sistema.

- O método `sendLastPublicComment` assume que o número de telefone e os comentários públicos são válidos e disponíveis. Caso contrário, ele lança exceções apropriadas.

## Vulnerabilidades

- **Dependência de dados externos:** O método depende de dados armazenados nos objetos `Case` e `CaseComment`. Se esses dados estiverem ausentes ou incorretos, o método falhará.
- **Exceções não tratadas:** Embora o método lance exceções, é importante garantir que elas sejam tratadas adequadamente no código que chama este método.
- **Segurança de dados:** Certifique-se de que as regras de compartilhamento e permissões de acesso estejam configuradas corretamente para evitar exposição de dados sensíveis.

Esta documentação detalhada deve ajudar qualquer desenvolvedor a entender e utilizar a classe `CaseMessageFromCommentController` de forma eficaz.