

Documentação do arquivo

SendMessageFromCaseCommentFlow.cls

Introdução

O arquivo `SendMessageFromCaseCommentFlow.cls` contém uma classe Apex que permite enviar mensagens via WhatsApp utilizando o serviço Atendechat. Ele é projetado para ser usado em fluxos do Salesforce, permitindo que comentários de casos sejam enviados como mensagens.

Descrição

A classe `SendMessageFromCaseCommentFlow` é uma classe pública com compartilhamento habilitado (`with sharing`), garantindo que as regras de segurança e compartilhamento do Salesforce sejam respeitadas. Ela possui um método invocável que pode ser chamado em fluxos do Salesforce para enviar mensagens do WhatsApp com base nos comentários de casos.

O método principal, `send`, verifica se o comentário está publicado e se o caso associado possui um número de telefone válido. Caso essas condições sejam atendidas, ele utiliza a classe `SendMessageAtendechat` para enviar a mensagem.

Estrutura

A estrutura do arquivo é simples e consiste em:

- Uma classe pública chamada `SendMessageFromCaseCommentFlow`.
- Um método estático invocável chamado `send`.

Dependências

A classe depende de:

- A classe `SendMessageAtendechat`, que é responsável por enviar mensagens via WhatsApp.
- O objeto padrão `Case` do Salesforce.
- O objeto padrão `CaseComment` do Salesforce.

Imports

Não há importações explícitas no código, pois o Apex não utiliza declarações de importação como em outras linguagens. No entanto, ele utiliza os objetos padrão do Salesforce (`Case` e `CaseComment`) e a classe personalizada `SendMessageAtendechat`.

Variáveis

O código não define variáveis globais ou de instância. As variáveis locais utilizadas são:

- `comments`: Lista de comentários de casos recebida como parâmetro.
- `comment`: Representa cada comentário individual na lista.
- `parentCase`: Representa o caso associado ao comentário.

Métodos

Método: send

Descrição:

Este método é invocável e pode ser chamado em fluxos do Salesforce. Ele recebe uma lista de comentários de casos e, para cada comentário publicado, verifica se o caso associado possui um número de telefone válido. Caso essas condições sejam atendidas, ele envia o corpo do comentário como uma mensagem via WhatsApp utilizando a classe `SendMessageAtendechat`.

Parâmetros:

- `comments`: Uma lista de objetos `CaseComment`.

Funcionamento:

1. Itera sobre a lista de comentários.
2. Ignora comentários que não estão publicados (`IsPublished == false`).
3. Recupera o caso associado ao comentário usando uma consulta SOQL.
4. Verifica se o caso possui um número de telefone válido (`SuppliedPhone`).

5. Chama o método `send` da classe `SendMessageAtendechat` para enviar a mensagem.

Exemplo de uso:

Este método pode ser usado em um fluxo do Salesforce para automatizar o envio de mensagens do WhatsApp com base em comentários de casos.

Exemplo

```
// Exemplo de uso em um fluxo do Salesforce
List<CaseComment> comments = [SELECT Id, ParentId, CommentBody, IsPublished FROM CaseComment WHERE IsPublished = true];
SendMessageFromCaseCommentFlow.send(comments);
```

Diagrama de Dependência

```
classDiagram
    class SendMessageFromCaseCommentFlow {
        +send(List~CaseComment~ comments)
    }
    class Case {
        +Id
        +SuppliedPhone
    }
    class CaseComment {
        +Id
        +ParentId
        +CommentBody
        +IsPublished
    }
    class SendMessageAtendechat {
        +send(String phone, String message)
    }

    SendMessageFromCaseCommentFlow --> CaseComment : Utiliza
    SendMessageFromCaseCommentFlow --> Case : Consulta
    SendMessageFromCaseCommentFlow --> SendMessageAtendechat : Chama
```

Notas

- A classe utiliza a anotação `@InvocableMethod`, que permite que o método seja chamado em fluxos do Salesforce.
- A consulta SOQL dentro do método `send` utiliza a cláusula `LIMIT 1` para garantir que apenas um registro seja retornado, mesmo que múltiplos casos correspondam ao `ParentId`.

Vulnerabilidades

- Consulta SOQL em loop:** A consulta SOQL é executada dentro de um loop `for`, o que pode levar a problemas de desempenho e atingir limites de governança do Salesforce. Uma abordagem mais eficiente seria usar uma consulta SOQL para buscar todos os casos associados aos comentários antes de iterar sobre eles.
- Validação de número de telefone:** O código apenas verifica se o campo `SuppliedPhone` não está em branco. Não há validação para garantir que o número de telefone seja válido ou esteja no formato correto.
- Dependência externa:** A funcionalidade depende da classe `SendMessageAtendechat`, que não está incluída no código fornecido. Qualquer problema nessa classe pode impactar o funcionamento do método `send`.