

Documentação do arquivo SearchClientController.cls

Introdução

O arquivo `SearchClientController.cls` contém uma classe Apex que é utilizada para realizar buscas de clientes no Salesforce. Ele foi criado para fornecer uma funcionalidade de pesquisa eficiente e cacheável, permitindo que os usuários encontrem clientes com base em seus nomes.

Descrição

A classe `SearchClientController` é uma classe Apex pública com compartilhamento habilitado (`with sharing`). Ela contém um método estático chamado `searchClients`, que é exposto para uso em componentes Lightning através da anotação `@AuraEnabled`. Este método realiza uma consulta SOQL para buscar registros de clientes (`Client__c`) cujo nome contenha uma sequência de caracteres fornecida como parâmetro.

Estrutura

A estrutura do arquivo é simples e consiste em:

- Uma classe pública chamada `SearchClientController`.
- Um método estático chamado `searchClients`.

Dependências

Este arquivo depende dos seguintes elementos:

- Objeto personalizado `Client__c` no Salesforce.
- Campos do objeto `Client__c`: `adress__c`, `complement__c`, `Email__c`, `Id`, `Name`, `number_adress__c`, `phone__c`.

Imports

Não há importações explícitas no código, pois o Apex não utiliza declarações de importação como outras linguagens de programação.

Variáveis

Variáveis Locais

- `lName`: Uma string que contém o nome formatado para a consulta SOQL. É gerada concatenando o caractere `%` antes e depois do parâmetro `aName`.

Parâmetros

- `aName`: Parâmetro de entrada do método `searchClients`. Representa o nome ou parte do nome do cliente que será pesquisado.

Métodos

`searchClients`

Descrição

Este método realiza uma consulta SOQL no objeto personalizado `Client__c` para buscar clientes cujo nome contenha a sequência de caracteres fornecida no parâmetro `aName`. Ele retorna uma lista de registros de clientes ordenados em ordem alfabética crescente.

Parâmetros

- aName (String): Nome ou parte do nome do cliente a ser pesquisado.

Retorno

- List<Client__c>: Uma lista de registros do objeto Client__c que correspondem ao critério de pesquisa.

Funcionalidade

1. Formata o parâmetro aName para ser usado na cláusula LIKE da consulta SOQL.
2. Executa a consulta SOQL para buscar os registros de clientes.
3. Retorna os resultados ordenados por nome em ordem crescente.

Código

```
@AuraEnabled(cacheable=true)
public static list<Client__c> searchClients(string aName){
    String lName = '%' + aName + '%';
    return [SELECT adress__c,complement__c,Email__c,Id,Name,number_adress__c,phone__c
            FROM Client__c
            WHERE Name LIKE :lName
            ORDER BY Name ASC NULLS FIRST];
}
```

Exemplo

Uso do método searchClients em um componente Lightning

```
import { LightningElement, wire } from 'lwc';
import searchClients from '@salesforce/apex/SearchClientController.searchClients';

export default class ClientSearchComponent extends LightningElement {
    searchTerm = '';
    clients = [];

    handleSearch(event) {
        this.searchTerm = event.target.value;
        searchClients({ aName: this.searchTerm })
            .then(result => {
                this.clients = result;
            })
            .catch(error => {
                console.error('Erro ao buscar clientes:', error);
            });
    }
}
```

Diagrama de Dependência

O diagrama abaixo ilustra a relação entre a classe SearchClientController e o objeto personalizado Client__c.

```
classDiagram
    class SearchClientController {
        +searchClients(aName: String): List<Client__c>
    }
    class Client__c {
        +adress__c
        +complement__c
        +Email__c
        +Id
        +Name
        +number_adress__c
        +phone__c
    }
    SearchClientController --> Client__c : Consulta SOQL
```

Notas

- A anotação `@AuraEnabled(cacheable=true)` permite que o método seja chamado de componentes Lightning e habilita o cache para melhorar o desempenho.
- O uso de `with sharing` garante que as regras de compartilhamento do Salesforce sejam respeitadas ao executar a consulta SOQL.

Vulnerabilidades

- **Injeção SOQL:** Embora o código utilize a vinculação de variáveis (`:lName`), é importante garantir que o parâmetro `aName` seja validado antes de ser passado para o método, para evitar possíveis problemas de segurança.
- **Performance:** Consultas que retornam muitos registros podem impactar o desempenho. É recomendável implementar paginação ou limitar o número de registros retornados.