

Documentação do arquivo CampingListController.cls

Introdução

O arquivo `CampingListController.cls` é uma classe Apex que serve como controlador para gerenciar itens de camping. Ele fornece métodos para recuperar e salvar itens de camping no banco de dados Salesforce. Essa classe é projetada para ser usada em conjunto com componentes Aura ou Lightning Web Components (LWC).

Descrição

A classe `CampingListController` contém dois métodos principais:

- `getItems()`: Recupera uma lista de itens de camping do banco de dados.
- `saveItem(Camping_Item__c item)`: Insere um novo item de camping no banco de dados.

Ambos os métodos são marcados com a anotação `@AuraEnabled`, permitindo que sejam acessados por componentes Aura ou LWC.

Estrutura

A estrutura da classe é simples e direta, contendo:

- Uma definição de classe pública.
- Dois métodos estáticos públicos com a anotação `@AuraEnabled`.

Dependências

A classe depende do objeto personalizado `Camping_Item__c`, que deve estar definido no Salesforce. Este objeto contém os seguintes campos:

- `id`: Identificador único do item.
- `name`: Nome do item.
- `price__c`: Preço do item.
- `Quantity__c`: Quantidade disponível.
- `Packed__c`: Indica se o item está embalado.

Imports

Não há importações explícitas na classe, pois o Apex não utiliza o conceito de importação como em outras linguagens. No entanto, a classe utiliza o objeto personalizado `Camping_Item__c`.

Variáveis

Variáveis Locais

- `CI`: Lista de objetos `Camping_Item__c` usada no método `getItems()`.

Métodos

`getItems()`

Descrição:
Este método recupera todos os registros do objeto `Camping_Item__c` do banco de dados. Ele retorna uma lista de itens com os campos `id`, `name`, `price__c`, `Quantity__c` e `Packed__c`.

Parâmetros:
Nenhum.

Retorno:
`List<Camping_Item__c>` - Uma lista de itens de camping.

Exemplo de uso:

```
List< Camping_Item__c > items = CampingListController.getItems();
```

saveItem(Camping_Item__c item)

Descrição:

Este método insere um novo registro no objeto Camping_Item__c. Ele recebe um objeto Camping_Item__c como parâmetro e o insere no banco de dados.

Parâmetros:

- item: Um objeto do tipo Camping_Item__c que será inserido no banco de dados.

Retorno:

Camping_Item__c - O objeto inserido.

Exemplo de uso:

```
Camping_Item__c newItem = new Camping_Item__c(Name='Barraca', Price__c=200, Quantity__c=1, Packed__c=false);
Camping_Item__c savedItem = CampingListController.saveItem(newItem);
```

Exemplo

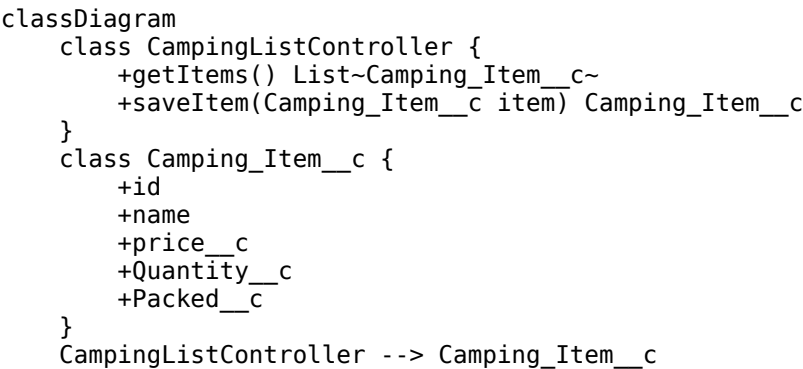
Aqui está um exemplo de como usar os métodos da classe CampingListController:

```
// Recuperar todos os itens de camping
List<Camping_Item__c> items = CampingListController.getItems();
System.debug('Itens de camping: ' + items);

// Salvar um novo item de camping
Camping_Item__c newItem = new Camping_Item__c(Name='Lanterna', Price__c=50, Quantity__c=2, Packed__c=true);
Camping_Item__c savedItem = CampingListController.saveItem(newItem);
System.debug('Item salvo: ' + savedItem);
```

Diagrama de Dependência

Abaixo está um diagrama de dependência que ilustra a relação entre a classe CampingListController e o objeto Camping_Item__c.



Notas

- Certifique-se de que o objeto personalizado Camping_Item__c e seus campos estejam configurados corretamente no Salesforce antes de usar esta classe.
- A anotação @AuraEnabled permite que os métodos sejam acessados por componentes Aura ou LWC, tornando esta classe útil para interfaces de usuário dinâmicas.

Vulnerabilidades

- **Inserção direta de dados:** O método saveItem insere diretamente o objeto recebido sem validação. Isso pode permitir a inserção de dados inválidos ou maliciosos. Recomenda-se adicionar validações antes de inserir o objeto.
- **Consulta sem filtros:** O método getItems retorna todos os registros do objeto Camping_Item__c sem aplicar filtros. Isso pode causar problemas de desempenho caso haja muitos registros no banco de dados.