

# Documentação do arquivo MyProfilePageController.cls

## Introdução

O arquivo `MyProfilePageController.cls` é uma classe Apex que gerencia a atualização dos detalhes de um usuário do portal. Ele garante que usuários convidados (guest users) nunca tenham acesso à página controlada por esta classe.

## Descrição

Esta classe é usada para manipular os dados do usuário logado no portal, permitindo que ele visualize e edite suas informações pessoais. A classe também inclui funcionalidades para salvar alterações, cancelar edições e alterar a senha do usuário. Além disso, ela implementa uma verificação para impedir que usuários convidados acessem a página.

## Estrutura

A classe possui variáveis privadas para armazenar o objeto `User` e um estado booleano que indica se o modo de edição está ativo. Ela também contém métodos públicos para interagir com os dados do usuário e gerenciar o estado da página.

## Dependências

- `User`: Objeto padrão do Salesforce que representa um usuário.
- `UserInfo`: Classe padrão do Salesforce usada para obter informações sobre o usuário logado.
- `ApexPages`: Classe padrão do Salesforce usada para adicionar mensagens de erro.
- `PageReference`: Classe padrão do Salesforce usada para redirecionar páginas.
- `DmlException`: Exceção lançada em operações DML (Data Manipulation Language).

## Imports

Não há importações explícitas, pois a classe utiliza objetos e classes padrão do Salesforce.

## Variáveis

### Variáveis privadas

- `User user`: Armazena os detalhes do usuário logado.
- `boolean isEdit`: Indica se o modo de edição está ativo.

## Métodos

### `getUser()`

- **Descrição:** Retorna o objeto `User` que contém os detalhes do usuário logado.
- **Retorno:** `User`

### `MyProfilePageController()`

- **Descrição:** Construtor da classe. Inicializa o objeto `User` com os detalhes do usuário logado. Verifica se o tipo de usuário é "GUEST" e lança uma exceção caso seja.
- **Exceção:** `NoAccessException` é lançada se o tipo de usuário for "GUEST".

### `getIsEdit()`

- **Descrição:** Retorna o estado do modo de edição.
- **Retorno:** `Boolean`

### `edit()`

- **Descrição:** Ativa o modo de edição.
- **Retorno:** Nenhum.

### `save()`

- **Descrição:** Salva as alterações feitas no objeto User. Caso ocorra um erro durante a operação DML, adiciona mensagens de erro à página.
- **Exceção:** DmlException é capturada e tratada.
- **Retorno:** Nenhum.

**changePassword()**

- **Descrição:** Redireciona o usuário para a página de alteração de senha.
- **Retorno:** PageReference

**cancel()**

- **Descrição:** Cancela o modo de edição e recarrega os dados do usuário logado.
- **Retorno:** Nenhum.

**Exemplo**

Abaixo está um exemplo de como usar a classe MyProfilePageController:

```
MyProfilePageController controller = new MyProfilePageController();
System.debug(controller.getUser()); // Exibe os detalhes do usuário logado
controller.edit(); // Ativa o modo de edição
controller.save(); // Salva as alterações feitas
controller.cancel(); // Cancela o modo de edição e recarrega os dados do usuário
PageReference changePasswordPage = controller.changePassword(); // Redireciona para a página de alteração de senha
```

**Diagrama de Dependência**

O diagrama abaixo ilustra as dependências entre os métodos e variáveis da classe:

```
classDiagram
    class MyProfilePageController {
        - User user
        - boolean isEdit
        + User getUser()
        + MyProfilePageController()
        + Boolean getIsEdit()
        + void edit()
        + void save()
        + PageReference changePassword()
        + void cancel()
    }
    class User {
        + id
        + email
        + username
        + usertype
        + communitynickname
        + timezonesidkey
        + languagelocalekey
        + firstname
        + lastname
        + phone
        + title
        + street
        + city
        + country
        + postalcode
        + state
        + localesidkey
        + mobilephone
        + extension
        + fax
        + contact.email
    }
    MyProfilePageController --> User
```

**Notas**

- A classe utiliza a palavra-chave with sharing para respeitar as regras de compartilhamento do Salesforce.
- A verificação de tipo de usuário garante que usuários convidados não possam acessar a página, aumentando a segurança.

**Vulnerabilidades**

- **Exceções não tratadas:** Embora a exceção `DmlException` seja capturada no método `save`, outras exceções podem ocorrer e não estão sendo tratadas.
- **Dependência de consultas SOQL:** A classe depende de consultas SOQL para carregar os dados do usuário. Isso pode ser um ponto de vulnerabilidade caso as consultas sejam mal otimizadas ou excedam os limites de governança do Salesforce.