# [Custom Page] Como inserir, atualizar e deletar um registro em um determinado Form estando em um Custom Page

## Para Inserir (Insert)

### Insert usando C#

```
public class ExemploMinhaClasseCsharp{

    public async Task<Guid> InserirRegistroAsync(string nome, string email){

        //troque o valor da variavel estruturaformularioidASerCriadoOuAtualizado para o Id do Form que você quer inserir um registro
        var estruturaformularioidASerCriadoOuAtualizado = Guid.Parse("0153b02c-d4d8-4747-9a69-45aafa113f81");

        //preencher a collection valuesToSave com os valores para o form que quer salvar
        var valuesToSave = new DataDictionary();
        valuesToSave.Add("name", nome);
        valuesToSave.Add("value1", email);

        var createdId = await CustomPageContext.SaveEntityAsync(estruturaformularioidASerCriadoOuAtualizado,
```

```
valuesToSave);

      return createdId;
    }



}

public ExemploMinhaClasseCsharp CreateInstanceExemploMinhaClasseCsharp(){
   var obj = CreateInstance<ExemploMinhaClasseCsharp>();
   return obj;
}
```

## Agora você precisa chamar esse metodo dentro do seu page<body> ou no partial cshtml.

```
@{
   var obj = CustomPageClass.CreateInstanceExemploMinhaClasseCsharp();
   var idCriado = await obj.InserirRegistroAsync("joao", "joao@joao.com");
}
idCriado: @idCriado
```

# Insert usando cshtml

```
@{
   //troque o valor da variavel estruturaformularioidASerCriadoOuAtualizado para o Id do Form que você quer inserir um registro
      var estruturaformularioidASerCriadoOuAtualizado = Guid.Parse("0153b02c-d4d8-4747-9a69-45aafa113f81");

      //preencher a collection valuesToSave com os valores para o form que quer salvar
      var valuesToSave = new DataDictionary();
      valuesToSave.Add("name", nome);
      valuesToSave.Add("value1", email);

      var createdId = await CustomPageContext.SaveEntityAsync(estruturaformularioidASerCriadoOuAtualizado, valuesToSave);
```

```
}
Id Form Criado: @createdId
```

# Para Atualizar (Update)

## Update usando C#

```
public class ExemploMinhaClasseCsharp{

      public async Task AtualizarRegistroAsync(Guid idRegistro, string nome, string email){

        //troque o valor da variavel estruturaformularioidASerCriadoOuAtualizado para o Id do Estrutura do Formulario que você quer atualizar um registro
        var estruturaformularioidASerCriadoOuAtualizado = Guid.Parse("0153b02c-d4d8-4747-9a69-45aafa113f81");

        //preencher a collection valuesToSave com os valores para o form que quer salvar
        var valuesToSave = new DataDictionary();
        valuesToSave.Add("id", idRegistro);
        valuesToSave.Add("name", nome);
        valuesToSave.Add("value1", email);

        await CustomPageContext.SaveEntityAsync(estruturaformularioidASerCriadoOuAtualizado, valuesToSave);

     }


}

public ExemploMinhaClasseCsharp CreateInstanceExemploMinhaClasseCsharp(){
   var obj = CreateInstance<ExemploMinhaClasseCsharp>();
   return obj;
}
```

Agora você precisa chamar esse metodo dentro do seu page<body> ou no partial cshtml.

```
@{
    var obj = CustomPageClass.CreateInstanceExemploMinhaClasseCsharp();

    var idRegistro = CustomPageContext.GetParam("idRegistro");
    await obj.AtualizarRegistroAsync(Guid.Parse(idRegistro), "joao", "joao@joao.com");
}
Form Atualizado
```

Uma das formas para Passar o Id do Registro que deve ser atualizado, você pode utilizar a função abaixo no javascript:

```
function loadPartialToUpdateRegister() {

    var cshtmlPartialName = "[coloque aqui o nome do partial cshtml que vc fez as regras de atualizar]";
    customPageContext.loadCsHtmlComponent(cshtmlPartialName,
        {
            placeholderId: 'result_cshtml', //parametro opcional, se o cshtml retornar algum html e vc quiser, vc pode colocar o retorno dentro de algum placeholder, div, etc..
            extraData: {
                idRegistro: $('#guidRegistro').val(), //supondo que vc tenha um campo (guidRegistro) na tela com o GUID preenchido ref ao id do registro que quer atualizar
            },
            onSuccess: function (result) {
                customPageContext.msg.success("Atualização com sucesso", "Confirmação");
            },
            onError: function (error) {
                customPageContext.msg.error("Atualização nao realizada", "Erro");
                console.log('error', error)
            }

        }
    )
}
```

# Update usando cshtml

```
@{
    var idRegistro = CustomPageContext.GetParam("idRegistro");

     //troque o valor da variavel estruturaformularioidASerCriadoOuAtualizado para o Id do Estrutura do Formulario que você quer atualizar um registro
    var estruturaformularioidASerCriadoOuAtualizado = Guid.Parse("0153b02c-d4d8-4747-9a69-45aafa113f81");

    //preencher a collection valuesToSave com os valores para o form que quer salvar
    var valuesToSave = new DataDictionary();
    valuesToSave.Add("id", idRegistro);
    valuesToSave.Add("name", nome);
    valuesToSave.Add("value1", email);

    await CustomPageContext.SaveEntityAsync(estruturaformularioidASerCriadoOuAtualizado, valuesToSave);

}
Form Atualizado
```

Uma das formas para Passar o Id do Registro que deve ser atualizado, você pode utilizar a função abaixo no javascript:

```
function loadPartialToUpdateRegister() {

   var cshtmlPartialName = "[coloque aqui o nome do partial cshtml que vc fez as regras de atualizar]";
   customPageContext.loadCsHtmlComponent(cshtmlPartialName,
      {
         placeholderId: 'result_cshtml', //parametro opcional, se o cshtml retornar algum html e vc quiser, vc pode colocar o retorno dentro de algum placeholder, div, etc..
         extraData: {
            idRegistro: $('#guidRegistro').val(), //supondo que vc tenha um campo (guidRegistro) na tela com o GUID preenchido ref ao id do registro que quer atualizar
         },
         onSuccess: function (result) {
            customPageContext.msg.success("Atualização com sucesso", "Confirmação");
```

```
        },
        onError: function (error) {
            customPageContext.msg.error("Atualização nao realizada", "Erro");
            console.log('error', error)
        }

    }
  )
}
```

# Para Deletar (Delete)

## Delete usando C#

```
public class ExemploMinhaClasseCsharp{

     public async Task DeletarRegistroAsync(Guid idRegistro){

       await CustomPageContext.DeleteAsync(idRegistro);

    }



}

public ExemploMinhaClasseCsharp CreateInstanceExemploMinhaClasseCsharp(){
   var obj = CreateInstance<ExemploMinhaClasseCsharp>();
   return obj;
}
```

Agora você precisa chamar esse metodo dentro do seu page<body> ou no partial cshtml.

```
@{
    var obj = CustomPageClass.CreateInstanceExemploMinhaClasseCsharp();

    var idRegistro = CustomPageContext.GetParam("idRegistro");
    await obj.DeletarRegistroAsync(Guid.Parse(idRegistro));
}
Form Deletado
```

Uma das formas para Passar o Id do Registro que deve ser atualizado, você pode utilizar a função abaixo no javascript:

```
function loadPartialToDeleteRegister() {

    var cshtmlPartialName = "[coloque aqui o nome do partial cshtml que vc fez as regras de deletar]";
    customPageContext.loadCsHtmlComponent(cshtmlPartialName,
        {
            placeholderId: 'result_cshtml', //parametro opcional, se o cshtml retornar algum html e vc quiser, vc pode colocar o retorno dentro de algum placeholder, div, etc..
            extraData: {
                idRegistro: $('#guidRegistro').val(), //supondo que vc tenha um campo (guidRegistro) na tela com o GUID preenchido ref ao id do registro que quer deletar
            },
            onSuccess: function (result) {
                customPageContext.msg.success("Removido com sucesso", "Confirmação");
            },
            onError: function (error) {
                customPageContext.msg.error("Remoção nao realizada", "Erro");
                console.log('error', error)
            }

        }
    )
}
```

# Delete usando cshtml

```
@{
    var idRegistro = CustomPageContext.GetParam("idRegistro");


    await CustomPageContext.DeleteAsync(idRegistro);


}
Form Atualizado
```

Uma das formas para Passar o Id do Registro que deve ser atualizado, você pode utilizar a função abaixo no javascript:

```
function loadPartialToDeleteRegister() {


   var cshtmlPartialName = "[coloque aqui o nome do partial cshtml que vc fez as regras de deletar]";
   customPageContext.loadCsHtmlComponent(cshtmlPartialName,
      {
         placeholderId: 'result_cshtml', //parametro opcional, se o cshtml retornar algum html e vc quiser, vc pode colocar o retorno dentro de algum placeholder, div, etc..
         extraData: {
            idRegistro: $('#guidRegistro').val(), //supondo que vc tenha um campo (guidRegistro) na tela com o GUID preenchido ref ao id do registro que quer deletar
         },
         onSuccess: function (result) {
            customPageContext.msg.success("Removido com sucesso", "Confirmação");
         },
         onError: function (error) {
            customPageContext.msg.error("Remoção nao realizada", "Erro");
            console.log('error', error)
         }


      }
   )
}
```

Revision #6
Created 4 March 2025 14:18:20 by agilityflow
Updated 5 March 2025 12:33:44 by agilityflow