



Documentação Técnica: Implementação da Criação de Caso (CS) no ServiceNow

Update Sets : CC - SCALA - PRB_Comments_to_CS

1. Introdução

Este documento descreve a implementação da funcionalidade de criação de Caso (CS) no ServiceNow, utilizando uma UI Page e um Script Include. Esta implementação garante que um CS seja criado apenas se não houver um CS já existente para o problema e se a empresa não for "SCALA DATA CENTERS S.A.".

2. Componentes

- **Script Include:** CreateCaseWithComments
- **UI Page:** create_cs_dialog
- **UI Action:** Create CS (Case)

3. Script Include

Nome: CreateCaseWithComments

```
var CreateCaseWithComments = Class.create();
```

```
CreateCaseWithComments.prototype = Object.extendObject(AbstractAjaxProcessor, {
```

```
  process: function() {
```

```
    var problemSysId = this.getParameter('sys_id');
```

```
    var assetStatus = this.getParameter('asset_status');
```

```
    var rootCauseInvestigation = this.getParameter('root_cause_investigation');
```

```
    var rcaAvailability = this.getParameter('rca_availability');
```

```
    var mitigationAction = this.getParameter('mitigation_action');
```

```
    // Verifique se o sys_id do problema é válido
```

```
    var problemGR = new GlideRecord('problem');
```

```
    if (!problemGR.get(problemSysId)) {
```

```
      return 'Problema não encontrado';
```



```
}

// Verifique se a empresa é SCALA DATA CENTERS S.A.
if (problemGR.company.name == 'SCALA DATA CENTERS S.A.') {
    return 'company_not_allowed';
}

// Criar o caso (CS)
var caseGR = new GlideRecord('sn_customerservice_case');
caseGR.initialize();

caseGR.short_description = problemGR.short_description;
caseGR.description = problemGR.description;

caseGR.account = problemGR.company; // Copiar valor do campo company para o campo account
na CS

caseGR.contact = problemGR.u_customer_account; // Copiar valor do campo u_customer_account
para o campo contact

caseGR.assignment_group = problemGR.assignment_group; // Copiar o grupo de atribuição
caseGR.assigned_to = problemGR.assigned_to; // Copiar atribuído a

var comments = "Status do Ativo: " + assetStatus + "\n\nInvestigação de causa Raiz: " +
rootCauseInvestigation + "\n\nData e Hora da Disponibilização do RCA: " + rcaAvailability + "\n\nAção
mitigatória adotada: " + mitigationAction;

caseGR.comments = comments;

var caseSysId = caseGR.insert();

if (!caseSysId) {
    return 'Erro ao criar o caso';
}

// Atualizar o campo Parent do problema com o sys_id do caso
problemGR.parent = caseSysId;
```



```
problemGR.comments = comments;  
  
problemGR.update();  
  
return caseSysId;  
}  
});
```

4. UI Page

Nome: create_cs_dialog

```
<j:jelly xmlns:j="jelly:core" xmlns:g="glide">  
  <style>  
    #dialog {  
      padding: 20px;  
      font-family: Arial, sans-serif;  
      font-size: 14px;  
      max-width: 500px;  
      margin: 0 auto;  
      border: 1px solid #ccc;  
      border-radius: 8px;  
      box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);  
      background-color: #fff;  
    }  
    #dialog h1 {  
      font-size: 18px;  
      margin-bottom: 20px;  
      text-align: center;  
      color: #333;  
    }  
  </style>  
</j:jelly>
```



```
}  
  
#dialog label {  
    display: block;  
    margin-bottom: 8px;  
    font-weight: bold;  
    color: #555;  
}  
  
#dialog input, #dialog textarea {  
    width: 100%;  
    margin-bottom: 15px;  
    padding: 10px;  
    border: 1px solid #ccc;  
    border-radius: 4px;  
    font-size: 14px;  
}  
  
#dialog textarea {  
    height: 100px;  
    resize: none;  
}  
  
#dialog .button-container {  
    text-align: right;  
}  
  
#dialog .button-container button {  
    padding: 10px 20px;  
    margin-left: 10px;  
    border: none;  
    border-radius: 4px;  
    cursor: pointer;  
    background-color: #007bff;
```



```
color: #fff;

font-size: 14px;

transition: background-color 0.3s ease;
}

#dialog .button-container button:hover {

    background-color: #0056b3;
}

#dialog .button-container button.secondary {

    background-color: #6c757d;
}

#dialog .button-container button.secondary:hover {

    background-color: #565e64;
}

#error_message {

    color: red;

    display: none;

    margin-bottom: 10px;
}

</style>

<div id="dialog">

    <h1>Create CS (Case)</h1>

    <form>

        <label for="asset_status">Status do Ativo:</label>

        <input type="text" id="asset_status" name="asset_status" />

        <label for="root_cause_investigation">Investigação de causa Raiz:</label>

        <input type="text" id="root_cause_investigation" name="root_cause_investigation" />
```



```
<label for="rca_availability">Data e Hora da Disponibilização do RCA:</label>

<input type="datetime-local" id="rca_availability" name="rca_availability" />


<label for="mitigation_action">Ação mitigatória adotada:</label>

<textarea id="mitigation_action" name="mitigation_action"></textarea>


<div id="error_message">Por favor, preencha todos os campos.</div>

<div class="button-container">

    <button type="button" onclick="submitCreateCS()">Enviar</button>

    <button type="button" class="secondary" onclick="closeDialog()">Fechar</button>

</div>

</form>

</div>


<script type="text/javascript">

function closeDialog() {

    GlideModal.get().destroy();

}


function submitCreateCS() {

    var assetStatus = document.getElementById('asset_status').value.trim();

    var rootCauseInvestigation = document.getElementById('root_cause_investigation').value.trim();

    var rcaAvailability = document.getElementById('rca_availability').value.trim();

    var mitigationAction = document.getElementById('mitigation_action').value.trim();

    var errorMessage = document.getElementById('error_message');

    if (!assetStatus || !rootCauseInvestigation || !rcaAvailability || !mitigationAction) {

        errorMessage.style.display = 'block';

        return;

    }

}
```



```
}

errorMessage.style.display = 'none';

var ga = new GlideAjax('CreateCaseWithComments');
ga.addParam('sys_id', g_form.getUniqueValue());
ga.addParam('asset_status', assetStatus);
ga.addParam('root_cause_investigation', rootCauseInvestigation);
ga.addParam('rca_availability', rcaAvailability);
ga.addParam('mitigation_action', mitigationAction);

ga.getXMLAnswer(function(response) {
    if (response) {
        if (response === 'company_not_allowed') {
            g_form.addErrorMessage('Não é permitido criar um CS para a empresa SCALA DATA CENTERS S.A. CSs só podem ser criados para clientes.');
        } else {
            var caseSysId = response;
            var gr = new GlideRecord('sn_customerservice_case');
            if (gr.get(caseSysId)) {
                var caseNumber = gr.getValue('number');
                var caseLink = '<a href="/sn_customerservice_case.do?sys_id=' + caseSysId + '"
target="_blank">' + caseNumber + '</a>';
                g_form.addInfoMessage('CS criado com sucesso: ' + caseLink);
            } else {
                g_form.addInfoMessage('CS criado com sucesso, mas não foi possível obter o número do
CS.');
            }
            g_form.reload();
        }
    }
}
```




```
    } else {  
        g_form.addErrorMessage("Erro ao criar CS. Por favor, tente novamente.");  
    }  
});  
  
closeDialog();  
}  
</script>  
</j:jelly>
```

UI Action

Nome: Create CS (Case)

- **Tabela:** Problem [problem]
- **Ordem:** 100
- **Nome da Ação:** create_cs
- **Ativo:** Sim
- **Mostrar Inserir:** Não
- **Mostrar Atualizar:** Sim
- **Cliente:** Sim
- **Compatível com List v2:** Sim
- **Compatível com List v3:** Não
- **OnClick:** openCSDialog();
- **Script:**

```
function openCSDialog() {  
    var problemSysId = g_form.getUniqueValue();  
    var parentValue = g_form.getValue('parent');  
    var companyName = g_form.getReference('company', function(company) {  
        if (parentValue) {  
            g_form.clearMessages();  
        }  
    });  
}
```




```
g_form.addErrorMessage('Um caso já está aberto para este problema: ' + parentValue);  
} else if (company.name == 'SCALA DATA CENTERS S.A.') {  
    g_form.clearMessages();  
    g_form.addErrorMessage('Não é permitido criar um CS para a empresa SCALA DATA CENTERS S.A.  
    CSs só podem ser criados para clientes.');
```



```
    } else {  
        var dialog = new GlideModal('create_cs_dialog');  
        dialog.setTitle('Create CS (Case)');  
        dialog.setPreference('sys_id', problemSysId);  
        dialog.render();  
    }  
});  
}  
  
openCSDialog();
```

6. Funcionamento

1. **Verificação de CS Existente:** Ao clicar na UI Action Create CS (Case), o script verifica se já existe um CS associado ao problema. Se existir, uma mensagem de erro é exibida.
2. **Verificação de Empresa:** O script também verifica se a empresa é "SCALA DATA CENTERS S.A.". Se for, uma mensagem de erro é exibida informando que CSs só podem ser criados para clientes.
3. **Criação do CS:** Se nenhuma das condições acima for atendida, o modal de criação de CS é aberto. O usuário preenche os campos e clica em "Enviar".
4. **Inserção do CS:** O Script Include CreateCaseWithComments é chamado para criar o CS e atualizar o problema com o CS recém-criado. O número do CS é exibido como um link na mensagem de sucesso.

7. Considerações Finais

Esta implementação garante que CSs só sejam criados em condições apropriadas e fornece feedback claro ao usuário. Ela utiliza componentes nativos do ServiceNow, garantindo fácil manutenção e extensibilidade futura.