



Event Listener Implementation for Protocol Deviation Summarization in Siebel CTMS

This document contains the event listener code that listens for a button click and triggers summary generation along with other UI updates. The code is part of a PR file and is associated with the **Clinical Protocol Management with Deviation Summarization** feature. This feature introduces a Generative AI-based solution in Siebel CTMS to automatically summarize protocol deviation records, reducing manual effort and improving clarity. It enhances protocol management efficiency by allowing users to review, edit, and apply concise summaries directly to the parent protocol records.

Event listener PR code

```
// Ensure SiebelJS Namespace and Definition
if (typeof SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR === "undefined") {
    SiebelJS.Namespace('SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR');

    define("siebel/custom/ProtocolDeviationDetailAppletCustomPR", ["siebel/phyrenderer"], function (PhyRenderer) {
        SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR = (function () {

            function ProtocolDeviationDetailAppletCustomPR(pm) {
                console.log('ProtocolDeviationDetailAppletCustomPR constructor called.');
```

SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR.superclass.constructor.call(this, pm);
 }

```

        SiebelJS.Extend(ProtocolDeviationDetailAppletCustomPR, SiebelAppFacade.PhysicalRenderer);

        // Init Method
        ProtocolDeviationDetailAppletCustomPR.prototype.Init = function () {
            console.log('ProtocolDeviationDetailAppletCustomPR Init method called.');
```

SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR.superclass.Init.call(this);
 };
 });
}

```

// BindEvents Method with jQuery Event Registration
ProtocolDeviationDetailAppletCustomPR.prototype.BindEvents = function () {
    console.log('ProtocolDeviationDetailAppletCustomPR BindEvents method called.');
```

SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR.superclass.BindEvents.call(this);

```

// Button Control (Ensure button ID is accurate for your configuration)
var buttonControl = $('#s_1_1_3_0_Ctrl'); // Replace with appropriate control ID

if (buttonControl.length) {
    console.log("Button control found. Setting up jQuery click event.");

    buttonControl.off("click").on("click", function () {
        console.log("Generate button clicked.");
        try {
            // Get the active applet and business component
            const applet = SiebelApp.S_App.GetActiveView().GetActiveApplet();
            if (!applet) {
                console.error("No active applet found.");
                return;
            }

            const busComp = applet.GetBusComp();
            if (!busComp) {
                console.error("No business component associated with the applet.");
                return;
            }

            // Perform query to retrieve records
            busComp.SetSearchSpec("Id", ""); // Adjust search spec if needed
            const recs = busComp.GetRawRecords();

            // Format records into a string
            const appendedString = recs

```

```

        .map((item, index) => `${index + 1}. ${item[16]}`)
        .join("\n");

        // Call the popup creation function
        createPopup(appendedString);
    } catch (error) {
        console.error("Error during the operation:", error);
    }
    });
} else {
    console.error("Button control not found.");
}
};

// Function to Create Popup
async function createPopup(appendedString) {
    // Fetch Chat Response
    const chatResponse = await fetchChatResponse(appendedString);

    // Popup and Overlay
    const overlay = $("

```

```

        transform: "translate(-50%, -50%)",
        width: "520px",
        padding: "26px",
        boxShadow: "0 6px 12px rgba(0, 0, 0, 0.3)",
        backgroundColor: "#fff",
        border: "1px solid #ccc",
        zIndex: "1000",
        fontFamily: "Arial, sans-serif"
    });

    // Add Content to Popup
    popup.append($("<h4>").text("AI Generated Summary").css({
        marginTop: "0",
        marginBottom: "10px",
        fontSize: "16px",
        textAlign: "left",
        color: "#333"
    }));

    const textarea = $("<textarea>").css({
        width: "100%",
        height: "195px",
        marginBottom: "10px",
        fontSize: "14px",
        resize: "none",
        border: "1px solid black",
        borderRadius: "0",
        padding: "8px",
        overflow: "auto",
        whiteSpace: "pre-wrap"
    }).val(chatResponse);

    popup.append(textarea);

```

```
// Add Buttons

const saveButton = $("<button>").text("Save").css({
    marginRight: "10px",
    padding: "8px 10px",
    border: "none",
    backgroundColor: "#385427",
    color: "#fff",
    fontSize: "15px",
    cursor: "pointer",
    borderRadius: "4px"
}).on("click", function () {
    // Save Logic
    const updatedText = textarea.val();
    console.log("Updated Text:", updatedText);
    // Remove Popup and Overlay
    overlay.remove();
    popup.remove();
});

const exitButton = $("<button>").text("Exit").css({
    padding: "8px 10px",
    border: "none",
    backgroundColor: "#c0533f",
    color: "#fff",
    fontSize: "15px",
    cursor: "pointer",
    borderRadius: "4px"
}).on("click", function () {
    // Close Logic
    overlay.remove();
    popup.remove();
});

popup.append(saveButton, exitButton);
```

```

    $("body").append(overlay, popup);
}

// Fetch Chat Response
async function fetchChatResponse(inputText) {
    try {
        const response = await fetch("http://127.0.0.1:5000/chat-response", {
            method: "POST",
            headers: {
                "Content-Type": "application/json"
            },
            body: JSON.stringify({ input: inputText })
        });

        if (response.ok) {
            const data = await response.json();
            return data.text.replace(/^(.*)$/, '$1'); // Clean up quotes
        } else {
            console.error("Failed to fetch chat response:", response.statusText);
            return inputText;
        }
    } catch (error) {
        console.error("Error during POST request:", error);
        return inputText;
    }
}

return ProtocolDeviationDetailAppletCustomPR;

})();

return "SiebelAppFacade.ProtocolDeviationDetailAppletCustomPR";
});
}

```



Connect with us

Call +1.800.ORACLE1 or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

 blogs.oracle.com

 facebook.com/oracle

 twitter.com/oracle

Copyright © 2025, Oracle and/or its affiliates. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.