readme.md

Genesys Cloud for Salesforce SDK CTI extensions

This repository contains an example that allows Genesys Cloud for Salesforce users to extend functionality through the Salesforce SDK CTI extensions.

Table of contents

Getting started

Prerequisites

- A version of the Genesys Cloud for Salesforce managed package installed in your Salesforce organization that supports CTI extensions.
- Record types in your Salesforce organization. For more information, see Tailor Business Processes to Different Users Using Record Types.

Background information

The Salesforce SDK contains a set of interfaces that allow developers to extend specific Salesforce Open CTI methods in the Genesys Cloud for Salesforce integration.

- `purecloud.CTIExtension.ClickToDial`
- `purecloud.CTIExtension.ScreenPop`
- `purecloud.CTIExtension.SaveLog`

To use the CTI extensions, implement one or more interfaces in an Apex class and select the Apex class in the Salesforce managed package settings.

Example

This example uses the `purecloud.CTIExtension.saveLog` interface to save an interaction log to a record type. The example performs the following actions:

- Gets an interaction and an interaction log from event data.
- Instantiates a new or an existing Task object based on the interaction log.
- If the interaction is outbound, sets a record type on the Task object.
- Saves the Task object and returns its ID.

To implement the example, follow these steps.

1. Create an Apex class.
2. Configure the extension points in the managed package.

Create an Apex class

In Salesforce, create an Apex class that implements `purecloud.CTIExtensions.saveLog`.

1. In Salesforce, open **Developer Console**.
2. Click **File** > **New** > **Apex Class**.
3. For the name of the class, enter **CustomCTIExtensions**.

4. Replace the content for the class with the following example code.

```
global with sharing class CustomCTIExtensions implements
purecloud.CTIExtension.SaveLog
{

    public String onSaveLog(String data) {
        Map<String, Object> eventData = (Map<String, Object>)
JSON.deserializeUntyped(data);
        Map<String, Object> interaction = ( Map<String, Object>)
eventData.get('interaction');
        Map<String, Object> callLog = ( Map<String, Object>)
eventData.get('callLog');
        String direction = (String)interaction.get('direction');
        String callLogId = '';

        Map<String, Schema.RecordTypeInfo> recordTypes =
Schema.SObjectType.Task.getRecordTypeInfosByName();
        Task t = (Task) JSON.deserialize(JSON.serialize(callLog),
Task.class);
        if (direction.toLowerCase() == 'outbound') {
            Schema.RecordTypeInfo selectedType = recordTypes.get('Sales
Record Type');
            t.recordTypeId =  selectedType.recordTypeId;
        }

        upsert t;
        callLogId = t.Id;

        return callLogId;
    }

}
```

5. Save the file.

Configure the extension points in the managed package

In Salesforce, configure the extension points in the managed package to use the Apex class that you created.

1. In Salesforce, click **Setup**.
2. Search for **Installed Packages**.
3. Under **Build**, click **Installed Packages**.
4. On the **Installed Packages** page, click **Configure** next to the Genesys Cloud for Salesforce package.

5. Under **Choose a Call Center**, select a version of the call center definition.
6. Under **Extension Point Settings**, select the Apex class that you created.
7. Click **Save**.

Additional information

- Extension points in Genesys Cloud for Salesforce (Genesys Cloud Resource Center).
- Use the extension points to customize saving interaction logs (Genesys Cloud Resource Center).
- Configure extension points (Genesys Cloud Resource Center).
- About Genesys Cloud for Salesforce (Genesys Cloud Resource Center).