

Open CTI 開発者ガイド

バージョン 38.0, Winter '17



本書の英語版と翻訳版で相違がある場合は英語版を優先するものとします。

© Copyright 2000–2016 salesforce.com, inc. All rights reserved. Salesforce およびその他の名称や商標は、salesforce.com, inc. の登録商標です。本ドキュメントに記載されたその他の商標は、各社に所有権があります。

目次

第 1 章: Open CTI の使用開始	1
UI が重要となる理由	2
Open CTI のメソッドパリティ	4
その他の Voice ソリューション	6
機能のカスタマイズ	6
Open CTI サポートポリシー	8
後方互換性	8
API サポート	8
第 2 章: コールセンター定義ファイル	9
コールセンター定義ファイルの形式	10
必須要素と必須属性	12
省略可能な要素と属性	14
<item> 要素の値の指定	15
コールセンター定義ファイルのサンプル	15
第 3 章: Open CTI の使用	18
Open CTI への接続	19
非同期コール	20
サンプル HTML ページ	20
Force.com Canvas の使用	26
ベストプラクティス	27
第 4 章: Lightning Experience のメソッド	28
Lightning Experience の disableClickToDial()	29
Lightning Experience の enableClickToDial()	30
Lightning Experience の getAppViewInfo()	32
Lightning Experience の getCallCenterSettings()	34
Lightning Experience の getSoftphoneLayout()	35
Lightning Experience の isSoftphonePanelVisible()	39
Lightning Experience の onClickToDial()	40
Lightning Experience の onNavigationChange()	42
Lightning Experience の refreshView()	43
Lightning Experience の runApex()	45
Lightning Experience の saveLog()	48
Lightning Experience の screenPop()	50
Lightning Experience の searchAndScreenPop()	53
Lightning Experience の setSoftphoneItemIcon()	56
Lightning Experience の setSoftphoneItemLabel()	58

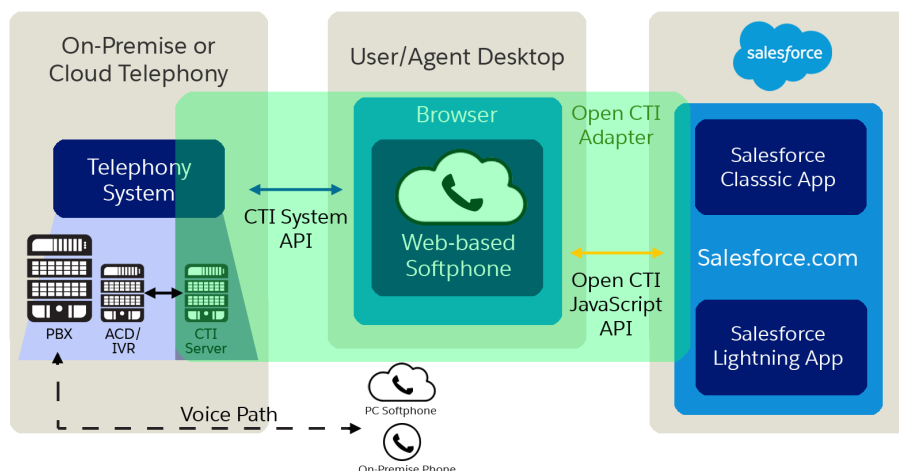
Lightning Experience の setSoftphonePanelHeight()	60
Lightning Experience の setSoftphonePanelIcon()	62
Lightning Experience の setSoftphonePanelLabel()	64
Lightning Experience の setSoftphonePanelVisibility()	66
Lightning Experience の setSoftphonePanelWidth()	67
Lightning Experience メソッドの一般的なエラーメッセージ	69
第 5 章: Salesforce Classic のメソッド	71
Salesforce アプリケーション操作のメソッド	71
Salesforce Classic の getPageInfo()	72
Salesforce Classic の isInConsole()	74
Salesforce Classic の isVisible()	75
Salesforce Classic の notifyInitializationComplete()	76
Salesforce Classic の onFocus()	77
Salesforce Classic の onObjectUpdate()	78
Salesforce Classic の refreshObject()	80
Salesforce Classic の refreshPage()	81
Salesforce Classic の refreshRelatedList()	82
Salesforce Classic の reloadFrame()	84
Salesforce Classic の runApex()	84
Salesforce Classic の saveLog()	86
Salesforce Classic の screenPop()	88
Salesforce Classic の searchAndGetScreenPopUrl()	89
Salesforce Classic の searchAndScreenPop()	91
Salesforce Classic の setVisible()	93
コンピュータテレフォニーインテグレーション (CTI) のメソッド	94
Salesforce Classic の disableClickToDial()	95
Salesforce Classic の enableClickToDial()	96
Salesforce Classic の getCallCenterSettings()	97
Salesforce Classic の getDirectoryNumbers()	98
Salesforce Classic の getSoftphoneLayout()	100
Salesforce Classic の onClickToDial()	103
Salesforce Classic の setSoftphoneHeight()	105
Salesforce Classic の setSoftphoneWidth()	106
第 6 章: その他のリソース	108
Open CTI 表記規則	108

第1章 Open CTI の使用開始

ブラウザベースの JavaScript API を使用して、サードパーティのコンピュータテレフォニーインテグレーション (CTI) システムを構築して Salesforce Call Center と統合します。

Salesforce で CTI 機能を表示するために、Open CTI はブラウザをクライアントとして使用します。Open CTI を使用すると、CTI アダプタをマシンにインストールしなくても Salesforce で直接ソフトフォンから電話をかけることができます。Open CTI 実装を開発したら、Salesforce Call Center を使用して Salesforce と統合できます。

Open CTI は、次のようにテレフォニーシステムに接続します。



エディション

使用可能なインターフェース: Salesforce Classic および Lightning Experience

使用可能なエディション: Professional Edition、Enterprise Edition、Performance Edition、Unlimited Edition、および Developer Edition

メモ: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

Open CTI では、次の操作ができます。

- CTI アダプタを使用せずに Salesforce と統合する CTI システムを構築する。
- Salesforce および Salesforce コンソールと完全に統合されたパーツとして機能するカスタマイズ可能なソフトフォン (通話管理ツール) を作成する。
- Mac、Linux、または Windows マシン上の Microsoft® Internet Explorer®、Mozilla® Firefox®、Apple® Safari®、または Google Chrome™ 向けの CTI などブラウザおよびプラットフォームにとらわれない CTI システムを提供する。

Open CTI を実装するには、CTI、JavaScript、Visualforce、Web サービス、ソフトウェア開発、Salesforce コンソール、Salesforce Call Center についての基本的な知識があると役立ちます。

Open CTI は、JavaScript ページでのみ使用できます。このガイドの例は JavaScript です。JavaScript で Open CTI を使用して、API コールおよびプロセスを埋め込むことができます。

UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較

Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。

Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性

2 つの API で提供されるメソッドには同じではないものもあります。Salesforce Classic メソッドには Lightning Experience では使用できないものや、名前が変更されているものがあります。

Open CTI とその他の Voice ソリューション

Open CTI はサードパーティの CTI システムと Salesforce を統合します。けれども、それ以前はどうなっていたのでしょうか。そして、Salesforce Voice との違いは何でしょうか。

Open CTI 機能のカスタマイズ

組織には、Open CTI 機能ではサポートされていない複雑なビジネスプロセスがあることがあります。心配は無用です。Force.com プラットフォームには、このような場合に高度な管理者や開発者がカスタム機能を実装できるさまざまな方法が搭載されています。

Open CTI サポートポリシー

Open CTI で機能強化が含まれるバージョンは、最新リリースのみです。

関連トピック:

[Salesforce ヘルプ: コールセンターの概要](#)

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: サポートされるブラウザ](#)

UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較

Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。

重要: これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

2 つの Open CTI API 間の違いは?

- API に異なる方法で接続します。

Salesforce Classic での接続

```
/support/api/38.0/interaction.js
```

Lightning Experience での接続`/support/api/38.0/lightning/opencti_min.js`

- メソッドの入力構文は異なります。

Salesforce Classic での接続

入力例

```
sampleMethod(var1, var2...)
```

Lightning Experience での接続

入力例

```
sampleMethod({  
  arg1 : value1,  
  arg2 : value2,  
  ...  
})
```

- この2つのAPIには同様のメソッドがありますが、いくつかのメソッドの動作は異なります。メソッドの入力および出力は異なる場合があります。

どの Open CTI API を使用するか?

API を切り替えることはできません。ユーザがユーザインターフェース間を切り替える場合、作業するユーザインターフェースによって CTI システムの動作や機能が異なる可能性があることをユーザが理解していることを確認してください。

Open CTI for Salesforce Classic は次の場合に使用します。

- Salesforce Classic アプリケーションでソフトフォンを使用して電話を発信する
- Salesforce コンソールでソフトフォンを使用して電話を発信する

Open CTI for Lightning Experience は次の場合に使用します。

- Lightning Experience アプリケーションでソフトフォンを使用して電話を発信する

設定に関する考慮事項はありますか?

Lightning Experience で電話を発信するには、次の操作を実行します。

- Lightning アプリケーションを作成し、Open CTI ソフトフォンオプションを追加します。
- コールセンター定義ファイルで、`reqSalesforceCompatibilityMode` 項目を *Lightning* または *Classic_and_Lightning* に設定する必要があります。

Open CTI for Lightning Experience は Lightning アプリケーションでのみ機能します。Salesforce Classic アプリケーションでは機能しません。Lightning Experience で Salesforce Classic アプリケーションを表示できますが、それらのアプリケーションは内部的には Classic アプリケーションのままです。アプリケーションが Lightning アプリケーションであるかどうかを確認するには、[設定] のアプリケーションマネージャを使用します。

Open CTI の実装が Lightning Experience と Salesforce コンソールで機能するようにするには、Open CTI for Salesforce Classic と Lightning Experience の両方を使用する固有の実装を開発します。コンソールのデザインは Lightning のように見える場合がありますが、使用できるのは Salesforce Classic でのみです。

関連トピック:

[Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性](#)

Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性

2つの API で提供されるメソッドには同じではないものもあります。Salesforce Classic メソッドには Lightning Experience では使用できないものや、名前が変更されているものがあります。

Salesforce Classic メソッド	Lightning Experience で使用可能?	Lightning Experience サポートに関する留意事項
<code>disableClickToDial()</code>	✓	Lightning の Open CTI で同じメソッド名を使用します。
<code>enableClickToDial()</code>	✓	Lightning の Open CTI で同じメソッド名を使用します。
<code>getCallCenterSettings()</code>	✓	Lightning の Open CTI で同じメソッド名を使用します。
<code>getDirectoryNumbers()</code>	✗	未サポート。
<code>getPageInfo()</code>	✓	Lightning メソッド <code>getAppViewInfo</code> の Open CTI で同じ機能が提供されます。
<code>getSoftphoneLayout()</code>	✓	Lightning の Open CTI で同じメソッド名を使用します。
<code>isInConsole()</code>	✗	Lightning Experience ではコンソールがサポートされていないため、使用できません。
<code>isVisible()</code>	✓	Lightning の <code>isSoftphonePanelVisible</code> の Open CTI で同じ機能が提供されます。
<code>notifyInitializationComplete()</code>	✗	未サポート。

Salesforce Classic メソッド	Lightning Experience で使用可能?	Lightning Experience サポートに関する留意事項
onClickToDial()	✓	Lightning の Open CTI で同じメソッド名を使用します。
onFocus()	✓	Lightning メソッド onNavigationChange の Open CTI で同じ機能が提供されます。
refreshPage()	✓	Lightning メソッド refreshView の Open CTI で同じ機能が提供されます。
refreshRelatedList()	✓	Lightning メソッド refreshView の Open CTI で同じ機能が提供されます。
reloadFrame()	✓	Lightning メソッド refreshView の Open CTI で同じ機能が提供されます。
runApex()	✓	Lightning の Open CTI で同じメソッド名を使用します。
saveLog()	✓	Lightning の Open CTI で同じメソッド名を使用します。
screenPop()	✓	Lightning の Open CTI で同じメソッド名を使用します。
searchAndGetScreenPopUrl()	✗	この機能を再現するには、Lightning の Open CTI で searchAndScreenPop を使用します。
searchAndScreenPop()	✓	Lightning の Open CTI で同じメソッド名を使用します。
setSoftphoneHeight()	✓	Lightning メソッド setSoftphonePanelHeight の Open CTI で同じ機能が提供されます。
setSoftphoneWidth()	✓	Lightning メソッド setSoftphonePanelWidth の Open CTI で同じ機能が提供されます。

Salesforce Classic メソッド	Lightning Experience で使用可能?	Lightning Experience サポートに関する留意事項
setVisible()	✓	Lightning メソッド setSoftphonePanelVisibility の Open CTI で同じ機能が提供されます。

関連トピック:

[UI が重要となる理由](#) — Open CTI for Salesforce Classic と Lightning Experience の比較

Open CTI とその他の Voice ソリューション

Open CTI はサードパーティの CTI システムと Salesforce を統合します。けれども、それ以前はどうなっていたのでしょうか。そして、Salesforce Voice との違いは何でしょうか。

Open CTI の前身は?

デスクトップ CTI (CTI Toolkit と呼ばれる) が Open CTI の前身です。デスクトップ CTI では、コールセンターの各ユーザのマシンにアダプタをインストールする必要がありました。Open CTI では、そのようなユーザ側のアダプタは過去のものになりました。

重要: Desktop CTI は廃止されるため、Open CTI に移行する必要があります。デスクトップ CTI 上に構築されたアダプタを使用している場合、そのアダプタは Spring '17 リリースをもって機能しなくなります。パートナーと協力して、Open CTI 実装を作成してください。

Salesforce Voice との違いは?

Salesforce Voice と Open CTI を混同しないでください。Salesforce Voice は、Salesforce から直接、番号のプロビジョニングと通話を行う方法を提供します。一方、すでにテレフォニーシステムがある場合には、既存のシステムと統合できる Open CTI が適しています。

関連トピック:

[CTI Toolkit の廃止に関する FAQ](#)

[Salesforce ヘルプ: Voice での電話の発着信](#)

Open CTI 機能のカスタマイズ

組織には、Open CTI 機能ではサポートされていない複雑なビジネスプロセスがあることがあります。心配は無用です。Force.com プラットフォームには、このような場合に高度な管理者や開発者がカスタム機能を実装できるさまざまな方法が搭載されています。

機能	説明
SOAP API	一度に1つのレコードタイプのみを処理し、トランザクション制御 (Savepoint の設定や変更のロールバックなど)を必要としない複合アプリケーションに機能を追加する場合、標準の SOAP API コールを使用します。 詳細については、『SOAP API 開発者ガイド』を参照してください。
Visualforce	Visualforce では、タグベースのマークアップ言語を使用して、開発者はより効果的にアプリケーションを開発したり、Salesforce のユーザインターフェースをカスタマイズしたりできます。Visualforce を使用して、次のことができます。 • ウィザードやその他のマルチステッププロセスの構築 • アプリケーションを介した独自のカスタムフローコントロールの作成 • 最適かつ効果的なアプリケーションの相互作用を目的とした、ナビゲーションパターンやデータ固有ルールの変換 詳細は、『Visualforce 開発者ガイド』を参照してください。
Salesforce コンソールインテグレーションツールキット	Salesforce コンソールインテグレーションツールキットは、Salesforce コンソールのカスタム機能を実装する場合に役立ちます。たとえば、Salesforce コンソールインテグレーションツールキットを使用して、Visualforce ページまたはサードパーティ製のコンテンツを Salesforce コンソールのタブとして表示できます。Salesforce コンソールインテグレーションツールキットは、ブラウザをクライアントとして使用し、コンソールにページを表示するための API です。 詳細は、『Salesforce コンソールインテグレーションツールキット開発者ガイド』を参照してください。
Apex	次のような場合に Apex を使用します。 • Web サービスを作成する • メールサービスを作成する • 複数のオブジェクトに複雑な検証を実行する • ワークフローでサポートされていない複雑なビジネスプロセスを作成する • カスタムトランザクションロジック (1つのレコードやオブジェクトだけでなく、トランザクション全体で発生するロジック)を作成する • レコードの保存などの別の操作にカスタムロジックを追加し、ユーザインターフェース、Visualforce ページ、SOAP API のいずれから操作が実行されても、ロジックが実行されるようにする 詳細は、『Apex 開発者ガイド』を参照してください。

Open CTI サポートポリシー

Open CTI で機能強化が含まれるバージョンは、最新リリースのみです。

以前のバージョンには、修正が含まれていない可能性があります。新しいバージョンがリリースされても、以前のバージョンを引き続き使用できます。

後方互換性

Salesforce では、Open CTI プラットフォームを使用する際に後方互換性を容易に確保できるよう努めています。

API サポート

Salesforce では、最初のリリース日から最低 3 年 Open CTI バージョンをサポートします。

後方互換性

Salesforce では、Open CTI プラットフォームを使用する際に後方互換性を容易に確保できるよう努めています。

新しい Salesforce リリースは、次の 2 つのコンポーネントで構成されています。

- Salesforce システムにある新しいリリースのプラットフォームソフトウェア
- 新しいバージョンの API

Open CTI は、各リリースの API バージョンと一致します。たとえば、SOAP API の現在のバージョンが 38.0 の場合、Open CTI もバージョン 38.0 が使用されます。

プラットフォームのリリースにわたって、各 Open CTI バージョンのサポートを維持しています。指定した Open CTI バージョンで機能するように作成されたアプリケーションが、その後のプラットフォームのリリースでも同じバージョンの Open CTI で継続して機能するという点では、Open CTI には後方互換性があります。

Salesforce では、あるバージョンの Open CTI に対応して作成されたアプリケーションがその後の Open CTI バージョンで機能することは保証されません。Open CTI の拡張は継続して行われているため、多くの場合、メソッド署名およびデータ表示の変更が必要になります。ただし、アプリケーションを新しい Open CTI バージョンに移行するために必要な変更が最小限ですむように、Open CTI のバージョン間の一貫性を保つように努めています。

たとえば、Summer '16 リリースに付属する Open CTI バージョン 37.0 を使用して作成されたアプリケーションは、Winter '17 リリースの Open CTI バージョン 37.0 およびそれ以降のリリースでも継続して機能します。ただし、同じアプリケーションが Open CTI バージョン 38.0 で機能するには、変更が必要になる可能性があります。

API サポート

Salesforce では、最初のリリース日から最低 3 年 Open CTI バージョンをサポートします。

Open CTI の品質とパフォーマンスを改善するために、リリースから 3 年を超えるバージョンはサポートを停止する可能性があります。

Open CTI バージョンのサポートが停止される予定がある場合、使用終了の通知が、バージョンのサポートが終了する最低 1 年前には送付されます。Salesforce は、使用できなくなる Open CTI バージョンを使用しているお客様に直接通知します。

第 2 章 コールセンター定義ファイル

コールセンター定義ファイルは、特定のソフトフォン向けの、Salesforce でコールセンターの定義に使用される一連の項目と値を指定します。Salesforce では、Salesforce CRM Call Center と複数の CTI システム業者とのインテグレーションをサポートするためにコールセンター定義ファイルを使用します。

Salesforce CRM Call Center のコールセンターには、ソフトフォンで特に動作するコールセンター定義ファイルが必要です。Open CTI でカスタムソフトフォンを開発する場合は、それをサポートするコールセンター定義ファイルを記述する必要があります。特定のソフトフォンに対するコールセンターの最初のインスタンスは、Salesforce にアダプタのコールセンター定義ファイルをインポートすることで定義する必要があります。その後のコールセンターは、最初のインポートで作成された元のコールセンターをコピーすることで作成できます。

組織でソフトフォンを変更したり、新規ソフトフォンを開発したりする場合は、元のソフトフォンのコールセンター定義ファイルをカスタマイズして、追加のコールセンターに必要な情報を含めるようにする必要があります。たとえば、バックアップサーバをサポートするシステムに対してソフトフォンを開発している場合、コールセンター定義ファイルにはそのバックアップサーバの IP アドレスとポート番号を含める必要があります。バックアップサーバのないシステムのソフトフォンに関連付けられているコールセンター定義ファイルには、これらの項目は必要ありません。

テキストエディタまたは XML エディタを使用して、コールセンター定義ファイルを定義します。

❗ 重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。コールセンター定義ファイルの `reqSalesforceCompatibilityMode` 項目では、使用するユーザインターフェース (Salesforce Classic、Lightning Experience、または両方) が識別されます。値を指定しない場合、*Classic* がデフォルトになります。この項目は省略可能ですが、Lightning Experience で電話を発信するには、*Lightning* または *Classic_and_Lightning* を指定する必要があります。

コールセンター定義ファイルの形式

コールセンター定義ファイルは、`callCenter`、`section`、`item` という 3 つの XML 要素で構成されます。

コールセンターの必須要素と必須属性

コールセンター定義ファイルでは、`<section>` 要素内に必須 `<item>` 要素を含める必要があります。

コールセンターの省略可能な要素と属性

コールセンター定義ファイルでは、`<section>` 要素内に省略可能な `<item>` 要素を含めることができます。

`<item>` 要素の値の指定

値をコールセンター定義ファイルに必ず指定する必要がある `reqInternalName` `<item>` を除き、`<item>` の値は、コールセンター定義ファイルで指定することも、定義ファイルのインポート後に Salesforce で指定することもできます。

コールセンター定義ファイルのサンプル

各コールセンター定義ファイルは異なります。この例では、Salesforce Classic および Lightning Experience を使用している組織のコールセンター定義ファイルを示します。

関連トピック:

[Salesforce ヘルプ: コールセンターの設定](#)

[Salesforce ヘルプ: コールセンターの作成](#)

コールセンター定義ファイルの形式

コールセンター定義ファイルは、callCenter、section、item という 3 つの XML 要素で構成されます。

次に、各要素のプロパティと属性について詳しく説明します。

callCenter

この要素は、1 つのコールセンターの電話システムの定義を表します。すべてのコールセンター定義ファイルには、少なくとも 1 つの `<callCenter>` 要素を含める必要があります。`<callCenter>` 要素は、1 つ以上の `<section>` 要素で構成されます。

section

この要素は、サーバ情報や発信番号など、関連するデータ項目のグルーピングを表します。Salesforce でコールセンターを編集すると、割り当てられたセクションごとに項目が整理されます。`<section>` 要素は、1 つの `<callCenter>` 要素に属し、1 つ以上の `<item>` 要素で構成されます。

属性:

名前	型	必須項目	説明
sortOrder	正の整数	必須	Salesforce でコールセンターを編集するときにセクションが表示される順序。たとえば、sortOrder="1" と指定されたセクションは、sortOrder="2" と指定されたセクションの直前に表示されます。 sortOrder の値は、負でない整数である必要があります。1 つのコールセンター定義内で数値をスキップすることはできません。たとえば、1 つのコールセンター定義ファイルに 3 つの <code><section></code> 要素がある場合、1 つの <code><section></code> 要素は sortOrder="0"、1 つの <code><section></code> 要素は sortOrder="1"、1 つの要素は <code><section></code> 要素は sortOrder="2" を持つ必要があります。
name	String	必須	Salesforce データベースで定義される、セクションの内部名。この値を使用して、カスタムアダプタまたはソフトフォンコードの作成時にセクションを参照できます。

名前	型	必須項目	説明
			名前には、空白や他の句読点を含まない英数字のみを使用する必要があります。それぞれの名前は40文字以内に制限されています。 req で始まる名前は、必須の Salesforce セクションのみのために予約されています(「コールセンターの必須要素と必須属性」を参照)。name 属性に使用できない他の予約語として、label、sortOrder、internalNameLabel、displayNameLabel があります。
label	String	省略可能	Salesforce で表示するときのセクション名。表示ラベルには、UTF-8 文字で構成された任意の文字列を使用できます。それぞれの名前は1000文字以内に制限されています。

item

この要素は、プライマリサーバのIPアドレスや国際電話の発信番号など、1つのコールセンター定義の1つの項目を表します。Salesforce でコールセンターを編集すると、各 `<item>` 要素は、それが属するセクションの下に表示されます。1つの `<item>` 要素に、複数の `<section>` 要素を指定できます。

属性:

名前	型	必須項目	説明
sortOrder	正の整数	必須	Salesforce でコールセンターを編集するときに項目が表示される順序。たとえば、sortOrder="1" と指定された項目は、sortOrder="2" と指定された項目の直前に表示されます。 sortOrder の値は、負でない整数である必要があります、1つのコールセンター定義内で数値をスキップすることはできません。たとえば、1つのコールセンター定義ファイルに3つの <code><item></code> 要素がある場合、1つの <code><item></code> 要素は sortOrder="0"、1つの <code><item></code> 要素は sortOrder="1"、1つの要素は <code><item></code> 要素は sortOrder="2" を持つ必要があります。
name	String	必須	Salesforce データベースで定義される、項目の内部名。この値を使用して、カスタムアダプタまたはソフトフォンコードの作成時に項目を参照できます。 名前には、空白や他の句読点を含まない英数字のみを使用する必要があります。それぞれの名前は40文字以内に制限されています。

名前	型	必須項目	説明
			req で始まる名前は、必須の Salesforce セクションのみのために予約されています (「 コールセンターの必須要素と必須属性 」を参照)。name 属性に使用できない他の予約語として、label、sortOrder、internalNameLabel、displayNameLabel があります。
label	String	省略可能	Salesforce で表示するときの項目名。表示ラベルには、UTF-8 文字で構成された任意の文字列を使用できます。それぞれの名前は 1,000 文字以内に制限されています。

関連トピック:

[Salesforce ヘルプ: コールセンター定義ファイル](#)

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

コールセンターの必須要素と必須属性

コールセンター定義ファイルでは、`<section>` 要素内に必須 `<item>` 要素を含める必要があります。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。コールセンター定義ファイルの `reqSalesforceCompatibilityMode` 項目では、使用するユーザインターフェース (Salesforce Classic、Lightning Experience、または両方) が識別されます。値を指定しない場合、`Classic` がデフォルトになります。この項目は省略可能ですが、Lightning Experience で電話を発信するには、`Lightning` または `Classic_and_Lightning` を指定する必要があります。

<code><item></code> の名前	説明	Salesforce Classic でサポート	Lightning Experience でサポート
<code>reqAdapterUrl</code>	CTI アダプタまたはソフトフォンをホストする場所を表します。次に例を示します。 <pre>http://localhost:11000</pre> 相対 URL は、Visualforce ページで有効です。次に例を示します。 <pre>: /apex/softphone</pre> Force.com Canvas アプリケーションを Open CTI に追加すると、そのアプリケーションを <code>reqAdapterUrl</code> より優先させることができます (指定されている場合)。  メモ: Lightning Experience 組織で実装するには、URL に <code>https</code> を使用します。	✓	✓

<item> の名前	説明	Salesforce Classic でサポート	Lightning Experience でサポート
reqCanvasApiName	コールセンターに追加される Force.com Canvas アプリケーションに関連付けられている API 名を表します。キャンバスアプリケーションを Open CTI に追加する場合は、必須です。	✓	✗ 未サポート
reqCanvasNamespace	コールセンターに追加される Force.com Canvas アプリケーションに関連付けられている名前空間を表します。キャンバスアプリケーションを Open CTI に追加する場合は、必須です。	✓	✗ 未サポート
reqDisplayName	Salesforce に表示されるコールセンターの名前を表します。sortOrder の値は 1 である必要があります。reqDisplayName の値は、最大長 1,000 文字の UTF-8 文字です。	✓	✓
reqInternalName	データベースでコールセンターを識別する一意の識別子を表します。sortOrder 値は 0 でなければならず、この値をコールセンター定義で指定する必要があります。reqInternalName の値には、空白や他の句読点を含まない 40 文字以内の英数字のみを使用できます。英字で始まり、組織で定義された他のすべてのコールセンターの reqInternalName で一意である必要があります。	✓	✓
reqSoftphoneHeight	Salesforce に表示されるソフトフォンの高さ (ピクセル単位) を表します。  メモ: Open CTI for Lightning Experience を使用している場合、240～700 の数値を入力します。値はピクセル (px) 単位です。	✓	✓
reqSoftphoneWidth	Salesforce に表示されるソフトフォンの幅 (ピクセル単位) を表します。  メモ: Open CTI for Lightning Experience を使用している場合、200～1240 の数値を入力します。値はピクセル (px) 単位です。	✓	✓
reqUseApi	コールセンターで Open CTI を使用しているか (true)、否か (false) を表します。	✓	✓

必要に応じて、このセクションに追加の `<item>` 要素を指定できます。

関連トピック:

[UI が重要となる理由](#) — [Open CTI for Salesforce Classic](#) と [Lightning Experience](#) の比較

コールセンターの省略可能な要素と属性

コールセンター定義ファイルでは、`<section>` 要素内に省略可能な `<item>` 要素を含めることができます。必須要素に加え、省略可能な要素を追加してソフトフォンを設定できます。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。コールセンター定義ファイルの `reqSalesforceCompatibilityMode` 項目では、使用するユーザインターフェース (Salesforce Classic、Lightning Experience、または両方) が識別されます。値を指定しない場合、`Classic` がデフォルトになります。この項目は省略可能ですが、Lightning Experience で電話を発信するには、`Lightning` または `Classic_and_Lightning` を指定する必要があります。

<code><item></code> の名前	説明	Salesforce Classic でサポート	Lightning Experience でサポート
<code>reqSalesforceCompatibilityMode</code>	ソフトフォンが表示されるかどうかを決定します。値を指定しない場合、<code>Classic</code> がデフォルトになります。  メモ: - Lightning Experience でソフトフォンを表示するには、<code>Lightning</code> を指定します。 - Salesforce Classic でソフトフォンを表示するには、<code>Classic</code> を指定します。 - 両方のユーザインターフェースでソフトフォンを表示するには、<code>Classic_and_Lightning</code> を指定します。	✓	✓
<code>reqStandbyUrl</code>	セカンダリソフトフォンをホストする場所を表します。スタンバイソフトフォンは、プライマリソフトフォンのタイムアウト期間が経過し、必要なタイムアウト期間内に Salesforce Classic の <code>notifyInitializationComplete()</code> メソッド がコールされていない場合に使用	✓	✗

<item> の名前	説明	Salesforce Classic でサ ポート	Lightning Experience で サポート
	用されます。スタンバイ URL を指定する場合は、reqTimeout 項目も指定する必要があります。		
reqTimeout	スタンバイ URL を使用してソフトフォンをホストするための経過時間 (ミリ秒単位) を表します。タイムアウト期間が経過するまでは、ソフトフォンが初期化中であることを示す読み込み中アイコンがソフトフォンに表示されます。必要なタイムアウトを指定する場合は、reqStandbyUrl 項目も指定する必要があります。	✓	✗

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

<item> 要素の値の指定

値をコールセンター定義ファイルに必ず指定する必要がある reqInternalName <item> を除き、<item> の値は、コールセンター定義ファイルで指定することも、定義ファイルのインポート後に Salesforce で指定することもできます。

コールセンター定義ファイルの <item> 要素に値を指定するには、<item> の開始タグおよび終了タグの間に値を配置します。たとえば、次のように使用します。

```
<item sortOrder="0" name="reqInternalName" label="Call Center Internal
Label">MyCallCenter</item>
```

reqInternalName <item> の値を MyCallCenter に設定します。reqInternalName の値以外の <item> の値は、コールセンター定義がインポートされた後に Salesforce で編集できます。

コールセンター定義ファイルのサンプル

各コールセンター定義ファイルは異なります。この例では、Salesforce Classic および Lightning Experience を使用している組織のコールセンター定義ファイルを示します。

❗ 重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。コールセンター定義ファイルの reqSalesforceCompatibilityMode 項目では、使用するユーザインターフェース (Salesforce Classic、Lightning Experience、または両方) が識別されます。値を指定しない場合、Classic がデフォルトになります。

す。この項目は省略可能ですが、Lightning Experience で電話を発信するには、*Lightning* または *Classic_and_Lightning* を指定する必要があります。

Salesforce Classic を使用している組織のサンプル XML

```
<!--
  All sections and items whose name value begins with "req" are
  required in a valid call center definition file. The sortOrder
  and label attributes can be changed for all required sections
  and items except reqGeneralInfo, reqInternalName, and
  reqDisplayName, in which only the label attribute can be altered.

  Note that the value for the reqInternalName item is limited to
  40 alphanumeric characters and must start with an alphabetic
  character. reqInternalName must be unique for all call centers
  that you define.
-->
<callCenter>
  <section sortOrder="0" name="reqGeneralInfo" label="General Information">
    <item sortOrder="0" name="reqInternalName" label="InternalName">DemoAdapter</item>
    <item sortOrder="1" name="reqDisplayName" label="Display Name">Demo Call Center
Adapter</item>
    <item sortOrder="2" name="reqAdapterUrl" label="CTI Adapter
URL">https://domain:port/softphone</item>
    <item sortOrder="3" name="reqUseApi" label="Use CTI API">true</item>
    <item sortOrder="4" name="reqSoftphoneHeight" label="Softphone Height">300</item>
    <item sortOrder="5" name="reqSoftphoneWidth" label="Softphone Width">500</item>
    <item sortOrder="6" name="reqSalesforceCompatabilityMode" label="Salesforce
Compatibility Mode">Classic</item>
  </section>
  <section sortOrder="1" name="reqDialingOptions" label="Dialing Options">
    <item sortOrder="0" name="reqOutsidePrefix" label="Outside Prefix">9</item>
    <item sortOrder="1" name="reqLongDistPrefix" label="Long Distance Prefix">1</item>
    <item sortOrder="2" name="reqInternationalPrefix" label="International Prefix">01</item>

  </section>
</callCenter>
```

Lightning Experience を使用している組織のサンプル XML

```
<callCenter>
  <section sortOrder="0" name="reqGeneralInfo" label="General Information">
    <item sortOrder="0" name="reqInternalName" label="InternalName">OpenCTI</item>
    <item sortOrder="1" name="reqDisplayName" label="Display Name">OpenCTI</item>
    <item sortOrder="2" name="reqAdapterUrl" label="CTI Adapter
URL">https://domain:port/softphone</item>
    <item sortOrder="3" name="reqUseApi" label="Use CTI API">true</item>
    <item sortOrder="4" name="reqSoftphoneHeight" label="Softphone Height">300</item>
    <item sortOrder="5" name="reqSoftphoneWidth" label="Softphone Width">500</item>
    <item sortOrder="6" name="reqSalesforceCompatibilityMode" label="Salesforce Compatibility
Mode">Lightning</item>
```

```
</section>
<section sortOrder="1" name="reqDialingOptions" label="Dialing Options">
  <item sortOrder="0" name="reqOutsidePrefix" label="Outside Prefix">9</item>
  <item sortOrder="1" name="reqLongDistPrefix" label="Long Distance Prefix">1</item>
  <item sortOrder="2" name="reqInternationalPrefix" label="International Prefix">01</item>

</section>
</callCenter>
```

関連トピック:

[UIが重要となる理由](#) — [Open CTI for Salesforce Classic](#) と [Lightning Experience](#) の比較

第 3 章 Open CTI の使用

Open CTI を使用することで、エージェントの効率を高め、ソフトフォンを設定し、その他の多くの作業を実行できます。

Open CTI では、次の操作ができます。

- ソフトフォンの高さまたは幅を設定する
- クリック-to-ダイヤルを有効化または無効化する
- コールセンター定義ファイルの設定を返す
- ユーザが Salesforce コンソールを表示しているかどうかを判別する
- Salesforce コンソールでソフトフォンを表示または非表示にする
- ページに関する情報を返す
- Salesforce で公開されている Apex クラスから Apex メソッドを実行する
- Salesforce のオブジェクトを保存または更新する
- Salesforce でキーワードを検索し、ソフトフォンレイアウトの定義に従って一致するレコードを画面ポップする

Open CTI 実装を開発する前に、Open CTI への接続方法を学習し、ベストプラクティスを確認しましょう。

Open CTI への接続

Open CTI を使用する JavaScript コードの冒頭部分では、ツールキットを JavaScript コードで使用できるようにする必要があります。このための構文は、Visualforce ページ、または Web テクノlogyを使用して開発されサーブドパーティドメインにあるページのどちらに JavaScript を埋め込んでいるかによって異なります。

Open CTI での非同期コール

Open CTI を使用すると、非同期コールを発行できます。非同期コールにより、クライアント側のプロセスは、サーバからのコールバックを待機せずに処理を続行できます。

Open CTI を使用したサンプル HTML ページ

Open CTI の実装はそれぞれ外観が異なります。この例では、HTML ページを使用して CTI 機能を Salesforce ユーザインターフェースに追加する方法を示します。

Force.com Canvas の使用

署名付き要求や OAuth 2.0 プロトコルなどの認証方式を必要とする外部アプリケーションに Open CTI を統合するには、Force.com Canvas を使用することをお勧めします。

ベストプラクティス

Open CTI を使用するときは、次のベストプラクティスに留意してください。

Open CTI への接続

Open CTI を使用する JavaScript コードの冒頭部分では、ツールキットを JavaScript コードで 사용할 수 있도록 하는必要があります。このための構文は、Visualforce ページ、または Web テクノロジを使用して開発されサードパーティドメインにあるページのどちらに JavaScript を埋め込んでいるかによって異なります。

 **ヒント:** Open CTI のバージョンは、URL に含まれています。

Visualforce ページの場合

Visualforce ページ、またはカスタム `onclick` JavaScript ボタン以外のソースの場合、Open CTI JavaScript ライブラリ ファイルを参照する `<script>` タグを指定します。

Salesforce Classic を使用している組織

```
<apex:page>
    <script src="/support/api/38.0/interaction.js" type="text/javascript"></script>
    ...
</apex:page>
```

Lightning Experience を使用している組織

```
<apex:page>
    <script src="/support/api/38.0/lightning/opencti_min.js"
type="text/javascript"></script>
    ...
</apex:page>
```

Visualforce では、相対パスを使用して `interaction.js` または `opencti_min.js` を含めることをお勧めします。

サードパーティドメインの場合

サードパーティドメインの場合、ツールキットを使用するには `interaction.js` または `opencti_min.js` への絶対 URL を指定する必要があります。組織のインスタンスを判断できない場合は、デフォルトインスタンスのツールキットライブラリにアクセスできます。デフォルトインスタンスの URL については、Salesforce にお問い合わせください。

Salesforce Classic を使用している組織

```
<script src="https://c.<yourInstance>.visual.force.com/support/api/38.0/interaction.js"
type="text/javascript"></script>
```

Lightning Experience を使用している組織

```
<script
src="https://c.<yourInstance>.visual.force.com/support/api/38.0/lightning/opencti_min.js"
type="text/javascript"></script>
```

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

Open CTI での非同期コール

Open CTI を使用すると、非同期コールを発行できます。非同期コールにより、クライアント側のプロセスは、サーバからのコールバックを待機せずに処理を続行できます。

非同期コールを発行するには、API コールに、コールバック関数と呼ばれる追加のパラメータを設定する必要があります。結果の準備ができると、サーバは結果を指定してコールバックメソッドを呼び出します。

非同期の構文:

Salesforce Classic では、次の手順を実行します。

```
method('arg1','arg2', ..., callback_method);
```

Lightning Experience では、次の手順を実行します。

```
method({callback : function})
```

例:

Salesforce Classic では、次の手順を実行します。

```
//Set softphone height
sforce.interaction.cti.setSoftphoneHeight(300, callback);
```

Lightning Experience では、次の手順を実行します。

```
//Disable clickToDial
sforce.opencti.disableClickToDial({callback: callback});
```

 **メモ:** コールの結果は、実行コンテキストによって異なります。たとえば、標準の Salesforce アプリケーションで `setSoftphoneWidth()` をコールしても影響はありませんが、Salesforce コンソールで `setSoftphoneWidth()` をコールするとソフトフォンの幅がサイズ変更されます。

Open CTI を使用したサンプル HTML ページ

Open CTI の実装はそれぞれ外観が異なります。この例では、HTML ページを使用して CTI 機能を Salesforce ユーザーインターフェースに追加する方法を示します。

この例では、Salesforce 組織にコールセンター定義ファイルをすでにインポートしたことを前提としています。

1. HTML ページを作成します。

2. 次のサンプルコードを切り取って HTML ページに貼り付けます。

このコードは、Open CTI の各種関数を示しています。

 **メモ:** Lightning Experience で電話をかけるには、まず Lightning アプリケーションを作成し、Open CTI ソフトフォンオプションを追加する必要があります。

Salesforce Classic のサンプルコード

```
<html>
<head>
  <!-- Imports Open CTI JavaScript library. Point to a valid Salesforce domain.
-->
  <script src="https://domain:port/support/api/38.0/interaction.js"></script>
  <script type="text/javascript">
    // Callback of API method: isInConsole
    var isInConsoleCallback = function (response) {
      // Returns true if method is executed in Salesforce console, false
      otherwise.

      if (response.result) {
        alert('Softphone is in Salesforce console.');
      }
      else {
        alert('Softphone is not in Salesforce console.');
      }
    };
    // Invokes API method: isInConsole
    function isInConsole() {
      sforce.interaction.isInConsole(isInConsoleCallback);
    }
    // Callback of API method: getCallCenterSettings
    var getCallCenterSettingsCallback = function (response) {
      // Result returns call center settings as a JSON string.
      if (response.result) {
        alert(response.result);
      }
      else {
        alert('Error retrieving call center settings ' +
response.error);
      }
    };
    // Invokes API method: getCallCenterSettings
    function getCallCenterSettings() {
      sforce.interaction.cti.getCallCenterSettings(getCallCenterSettingsCallback);
    }
    // Callback of API method: setSoftphoneHeight
    var setSoftphoneHeightCallback = function (response) {
      // Returns true if SoftPhone height was set successfully, false
      otherwise.

      if (response.result) {
        alert('Setting softphone height to 300px was successful.');
      }
      else {
```

```

        alert('Setting softphone height failed.');
```

```

    }
};
// Invokes setSoftphoneHeight API method.
function setSoftphoneHeight() {
    sforce.interaction.cti.setSoftphoneHeight(300,
setSoftphoneHeightCallback);
}
// Callback of API method: getPageInfo
var getPageInfoCallback = function (response) {
    if (response.result) {
        alert(response.result);
    }
    else {
        alert('Error occured while trying to get page info: ' +
response.error);
    }
}
// Invokes API method getPageInfo
function getPageInfo() {
    sforce.interaction.getPageInfo(getPageInfoCallback);
}
</script>
</head>
<body>
    <button onclick="isInConsole();">isInConsole</button><br/>
    <button onclick="getCallCenterSettings();">getCallCenterSettings</button><br/>

    <button onclick="setSoftphoneHeight();">setSoftphoneHeight(300)</button><br/>
    <button onclick="getPageInfo();">getPageInfo</button>
</body>
</html>

```

Lightning Experience のサンプルコード

```

<apex:page >
    <!-- Begin Default Content -->
    <h1>Congratulations!</h1>
    This is your sample page.
    <!-- End Default Content -->
</html>
<head>
    <!-- Imports Open CTI JavaScript library. Point to a valid Salesforce domain.
-->
    <script src="https://domain:port/support/api/38.0/opencti_min.js"></script>
    <script type="text/javascript">
        // Callback of API method: setSoftphonePanelHeight
        var setSoftphonePanelHeightCallback = function(response) {
            // Returns true if setSoftphonePanelHeight method is executed successfully,
            false otherwise
            if (response.result) {
                alert('setSoftphonePanelHeight is successfully executed.');
```

```

    }
};
// Invokes API method: setSoftphonePanelHeight
function setSoftphonePanelHeight() {
    sforce.opencti.setSoftphonePanelHeight({
        heightPX: 500,
        callback: setSoftphonePanelHeightCallback
    });
}
// Callback of API method: setSoftphonePanelWidth
var setSoftphonePanelWidthCallback = function(response) {
    // Returns true if setSoftphonePanelWidth method is executed successfully,
    false otherwise
    if (response.result) {
        alert('setSoftphonePanelWidth is successfully executed.');
```

```

    }
  };
  // Invokes API method: setSoftphoneItemLabel
  function setSoftphoneItemLabel() {
    sforce.opencti.setSoftphoneItemLabel({
      Label: 'MySoftphone',
      callback: setSoftphoneItemLabelCallback
    });
  }
</script>
</head>
<body>
  <button
onclick="setSoftphonePanelHeight();">setSoftphonePanelHeight({heightPX:500})</button><br/>

  <button
onclick="setSoftphonePanelWidth();">setSoftphonePanelWidth({widthPX:500})</button><br/>

  <button
onclick="setSoftphoneItemIcon();">setSoftphoneItemIcon({key:'call'})</button><br/>
  <button
onclick="setSoftphoneItemLabel();">setSoftphoneItemLabel({label:'MySoftphone'})</button>
</body>
</html>
</apex:page>

```

HTML ページを作成した後、コールセンター定義ファイルに URL を追加します。ソフトフォンは、Salesforce Classic では左側、Salesforce コンソールまたは Lightning Experience ではフッターに表示されます。

Salesforce Classic でのサンプル HTML ページの出力

Case Number	Subject	Date/Time Opened	Priority
00001000	Sample Case: Our Widgets have not been delivered.	1/25/2016 3:21 PM	High
00001003	-	1/25/2016 3:36 PM	Medium
00001009	-	4/8/2016 9:59 AM	Medium
00001008	-	4/8/2016 9:57 AM	Medium

Salesforce コンソールでのサンプル HTML ページの出力

The screenshot shows the Salesforce Classic interface. At the top is a blue header with the Salesforce logo and a search bar. Below it is a navigation bar with 'Cases' selected. The main area displays a table of cases under the 'My Open Cases' view. The table has columns for ACTION, CASE NUMBER, CONTACT NAME, SUBJECT, STATUS, PRIORITY, DATE/TIME OPEN, and CASE OWNER. A right-click context menu is open over the first case (00001000), showing several actions: 'isInConsole', 'getCallCenterSettings', 'setSoftphoneHeight(300)', and 'getPageInfo'. The bottom of the console shows a 'Phone' button.

ACTION	CASE NUMBER	CONTACT NAME	SUBJECT	STATUS	PRIORITY	DATE/TIME OPEN	CASE OWNER
[icon]	00001000	Amos, Jon	Sample Case: Our...	Escalated	High	1/25/2016 3:2...	TUser
[icon]	00001001	Stamos, Edward	Sample Case 2: Th...	New	High	1/25/2016 3:2...	TUser
[icon]	00001002	Stamos, Edward	Sample Case 3: C...	On Hold	Low	1/25/2016 3:21 P...	TUser
[icon]	00001003	-	-	New	Medium	1/25/2016 3:36 P...	TUser
[icon]	00001004	-	-	New	Medium	4/5/2016 10:11 P...	TUser
[icon]	00001005	-	-	New	-	-	-
[icon]	00001006	-	-	New	-	-	-
[icon]	00001007	-	-	New	-	-	-
[icon]	00001008	-	-	New	-	-	-
[icon]	00001009	-	-	New	-	-	-

Lightning Experience でのサンプル HTML ページの出力

The screenshot shows the Salesforce Lightning Experience interface. At the top is a navigation bar with 'testapp' and various tabs like 'Accounts', 'Assets', 'Calendar', 'Campaigns', 'Cases', 'Chatter', 'Contacts', and 'Contracts'. The 'Cases' tab is selected. Below it is a header for 'All Open Cases' with a 'New' button. The main area displays a table of cases. A right-click context menu is open over the first case (00001000), showing several actions: 'setSoftphonePanelHeight', 'setSoftphonePanelWidth', 'setSoftphoneItemIcon', and 'setSoftphoneItemLabel'. The bottom of the console shows a 'Phone' button.

CASE NUMBER	CONTACT NAME	SUBJECT	STATUS	PRIORITY
1 00001000	Jon Amos	Sample Case: Our Widgets...	Escalated	High
2 00001001	Edward Stamos	Sample Case 2: The widget...	New	High
		Sample Case 3: Cannot tra...	On Hold	Low
			New	Medium
			New	Medium
			New	Medium
			New	Medium
			New	Medium

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

Force.com Canvas の使用

署名付き要求や OAuth 2.0 プロトコルなどの認証方式を必要とする外部アプリケーションに Open CTI を統合するには、Force.com Canvas を使用することをお勧めします。

❗ 重要: Open CTI for Lightning Experience では Force.com Canvas はサポートされていません。

Force.com Canvas と Open CTI は似ています。これらはツールと JavaScript API のセットで、開発者がサードパーティシステムを Salesforce に追加するために使用できます。ただし、Force.com Canvas の利点の 1 つとして、認証方式を選択できるという点があります。

📌 メモ: キャンバスアプリケーションを Salesforce コンソールに表示するには、コンソールにカスタムコンソールコンポーネントとしてそれを追加する必要があります。詳細は、『*Canvas 開発者ガイド*』を参照してください。

キャンバスアプリケーションの開発時に Open CTI の機能を含めるには、次の操作を実行します。

1. `index.jsp` に Open CTI API を含めます。
2. `Sfdc.canvas.client.signedrequest()` をコールし、コンソールインテグレーションツールキット API に必要な署名付き要求を保存します。たとえば、Force.com Canvas の認証方式が署名付き要求の場合は、次を実行します。

```
Sfdc.canvas.client.signedrequest('<%=signedRequest%>')
```

Force.com Canvas の認証方式が OAuth の場合は、『*Canvas 開発者ガイド*』に記載されているように、コンテキストの取得に使用されるコールバック関数で次を実行します。

```
Sfdc.canvas.client.signedrequest(msg)
```

Open CTI およびキャンバスアプリケーションを使用する場合は、次の点を考慮してください。

- Open CTI API スクリプトは署名付き要求によって異なるため、`Sfdc.canvas.client.signedrequest()` へのコールの実行後に追加する必要があります。スクリプトを動的に読み込むことをお勧めします。
- キャンバスのサイドバーコンポーネントに関連付けられているレコードのエンティティ ID を取得するには、次を実行します。

```
// Get signedRequest
var signedRequest = Sfdc.canvas.client.signedrequest();
var parsedRequest = JSON.parse(signedRequest);
// get the entity Id that is associated with this canvas sidebar component.
var entityId = parsedRequest.context.environment.parameters.entityId;
```

- OAuth の `entityId` を取得するには、次を実行します。

```
var entityId = msg.payload.environment.parameters.entityId;
```

`msg.payload` の取得方法の例については、『*Canvas 開発者ガイド*』を参照してください。

関連トピック:

[Salesforce Canvas 開発者ガイド: キャンバスアプリケーションでのコンテキストの取得](#)

[Salesforce ヘルプ: アプリケーションへのコンソールコンポーネントの追加](#)

ベストプラクティス

Open CTI を使用するときは、次のベストプラクティスに留意してください。

- Open CTI に含まれるメソッドの多くは非同期で、コールバックメソッドを使用して結果を返すため、各メソッドのドキュメントを参照してそれぞれの応答の情報を理解することをお勧めします。
- Open CTI で生成されるエラーは通常、JavaScript 処理を停止しない方法で発行されます。コードをデバッグしやすいように、ブラウザに組み込まれた開発者ツールを使用して JavaScript コンソールを監視することをお勧めします。
- Visualforce を使用して Salesforce コンソールのサイドバーをカスタマイズ、拡張、または統合する場合は、オンラインヘルプでコンソールコンポーネントに関する情報を確認してください。

関連トピック:

[Salesforce ヘルプ: コンソールコンポーネント](#)

第 4 章 Lightning Experience のメソッド

組織で Lightning Experience を使用している場合、Lightning Experience で機能するメソッドを使用します。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

Lightning Experience の disableClickToDial()
Lightning Experience の enableClickToDial()
Lightning Experience の getAppViewInfo()
Lightning Experience の getCallCenterSettings()
Lightning Experience の getSoftphoneLayout()
Lightning Experience の isSoftphonePanelVisible()
Lightning Experience の onClickToDial()
Lightning Experience の onNavigationChange()
Lightning Experience の refreshView()
Lightning Experience の runApex()
Lightning Experience の saveLog()
Lightning Experience の screenPop()
Lightning Experience の searchAndScreenPop()
Lightning Experience の setSoftphoneItemIcon()
Lightning Experience の setSoftphoneItemLabel()
Lightning Experience の setSoftphonePanelHeight()
Lightning Experience の setSoftphonePanelIcon()
Lightning Experience の setSoftphonePanelLabel()
Lightning Experience の setSoftphonePanelVisibility()
Lightning Experience の setSoftphonePanelWidth()

[Lightning Experience メソッドの一般的なエラーメッセージ](#)

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

[Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性](#)

Lightning Experience の disableClickToDial()

使用方法

クリック-to-ダイヤルを無効にします。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.disableClickToDial({
  callback: function //Optional
})
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function disableClickToDial() {
        sforce.opencti.disableClickToDial({callback: callback});
      }
    </script>
  </head>
```

```
<body>
  <button onclick="disableClickToDial();">disableClickToDial()</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の enableClickToDial()

使用方法

クリック-to-ダイアルを有効にします。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.enableClickToDial({
  callback: function //Optional
})
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function enableClickToDial() {
        sforce.opencti.enableClickToDial({callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="enableClickToDial();">enableClickToDial()</button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の getAppViewInfo()

使用方法

現在のアプリケーションビューに関する情報を返します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.getAppViewInfo({
    callback: function
});
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function getAppViewInfo() {
        sforce.opencti.getAppViewInfo({callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="getAppViewInfo();">getAppViewInfo()</button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
<code>success</code>	<code>boolean</code>	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
<code>returnValue</code>	<code>object</code>	現在のアプリケーションビューの URL を返し、該当するレコード ID、レコード名、およびオブジェクト種別が含まれます。次に例を示します。 <pre>{ "url": "https://<i>yourInstance</i>.salesforce.com/one/one.app#/sObject/500D0000003tcchIAA / view ? t = 1459806689555 ", "recordId": "001x0000003DGQR", "recordName": "Acme", "objectType": "Account" }</pre> 個人取引先オブジェクトでこの API メソッドを呼び出すと、次の追加情報が返されます。 <code>accountId</code> または <code>contactId</code> — 関連する取引先または取引先責任者の ID。 <code>personAccount</code> — 組織で個人取引先が有効になっている場合は <code>true</code>、それ以外の場合は <code>false</code>。 次に例を示します。 <pre>{ "url": "http://<i>yourInstance</i>.salesforce.com/001x0000003DGQR", "recordId": "001x0000003DGQR", "recordName": "Acme Person Account", "objectType": "Account", "contactId": "003D000000QOMgg", "personAccount": true }</pre>
<code>error</code>	<code>array</code>	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の getCallCenterSettings ()

使用方法

現在のユーザに関連するコールセンターの設定を返します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.getCallCenterSettings({
  callback: function
});
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function getCallCenterSettings() {
        sforce.opencti.getCallCenterSettings({callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="getCallCenterSettings();">getCallCenterSettings() </button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	API コールが正常に行われた場合、コールセンターの設定が返されます。 <pre> { "/displayNameLabel": "Display Name Label", "/internalNameLabel": "Internal Name Label", "/reqDialingOptions/reqInternationalPrefix": "01" "/reqDialingOptions/reqLongDistPrefix": "1", "/reqDialingOptions/reqOutsidePrefix": "9", "/reqGeneralInfo/reqAdapterUrl": "http://localhost:8080", "/reqGeneralInfo/reqDescription": "Test Call Center", "/reqGeneralInfo/reqDisplayName": "Test Call Center", "/reqGeneralInfo/reqInternalName": "TestCallCenter", "/reqGeneralInfo/reqSalesforceCompatibilityMode": "Lightning", "/reqGeneralInfo/reqSoftphoneHeight": "500", "/reqGeneralInfo/reqSoftphoneWidth": "400", "/reqGeneralInfo/reqStandbyUrl": "http://localhost:8080/", "/reqGeneralInfo/reqUseApi": "true" } </pre>
error	array	API コールが失敗した場合、 <code>null</code> が返されます。 API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の getSoftphoneLayout ()

使用方法

現在のユーザのソフトフォンレイアウトを返します。このメソッドはAPIバージョン38.0以降で使用できます。

 **メモ:** Open CTI for Lightning Experience メソッド `screenPop ()` および `searchAndScreenPop ()` では、`screenPopSettings` はサポートされていません。

構文

```
sforce.opencti.getSoftphoneLayout({
    callback: function
});
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
<head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
        var callback = function(response) {
            if (response.success) {
                alert(response.returnValue);
            } else {
                console.error(response.errors);
                alert(
                    'Something went wrong. Please check error information in developer console.'
                );
            }
        };

        function getSoftphoneLayout() {
            sforce.opencti.getSoftphoneLayout({
                callback: callback
            });
        }
    </script>
</head>
<body>
    <button onclick="getSoftphoneLayout();">Get Softphone Layout</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API コールが正常に行われた場合、値 <code>true</code> が返されて <code>returnValue</code> でソフトフォンレイアウト定義が返され、それ以外の場合は <code>false</code> が返されます。
returnValue	object	API コールが正常に行われた場合、ソフトフォンレイアウト定義が返されます。API コールが失敗した場合、<code>null</code> が返されます。 返されたオブジェクトには、各通話種別を表す 3 つの要素が含まれます。 • <code>"Internal"</code> • <code>"Inbound"</code> • <code>"Outbound"</code> 各通話種別には 3 つのサブセクションが含まれます。 • <code>"callRelatedFields"</code> — 表示対象として選択された通話関連項目の配列。使用できる値は、<code>"ANI"</code>、<code>"DNIS"</code>、<code>"SEGMENT"</code>、<code>"QUEUE"</code> です。 • <code>"objects"</code> — 表示対象として選択された Salesforce オブジェクトのセット。各オブジェクトから表示対象として選択された [項目表示ラベル] および [項目名] (API 参照名) も含まれます。 • <code>"screenPopSettings"</code> — このオブジェクトには、<code>"ExistingWindow"</code> または <code>"NewWindow"</code> のいずれかの値を持つ <code>"screenPopsOpenWithin"</code> 項目が含まれます。 また、各画面ポップの一致項目、<code>"NoMatch"</code>、<code>"SingleMatch"</code>、<code>"MultipleMatches"</code> の設定も含まれます。各一致項目には、対応する <code>"screenPopType"</code> 項目が格納されます。また、<code>"screenPopData"</code> 項目が格納される場合もあります。 - <code>"screenPopType"</code> に <code>"PopToEntity"</code> の値が指定されている場合、<code>"screenPopData"</code> には対象オブジェクトの名前が含まれます。 - <code>"screenPopType"</code> に <code>"PopToVisualforce"</code> の値が指定されている場合、<code>"screenPopData"</code> には対象 Visualforce ページの名前が含まれます。 - <code>"screenPopType"</code> に <code>"PopToSearch"</code> の値が指定されている場合、<code>"screenPopData"</code> 項目は存在しません。

次に、`returnValue` の例を示します。

```
{
  "Internal" : {
    "callRelatedFields" : [
      "ANI",
      "DNIS",
```

名前	型	説明
		<pre>] "objects" : { "User" : [{ "displayName" : "Name", "apiName" : "Name" }], }, "screenPopSettings" : {} }, "Inbound" : { "callRelatedFields" : ["ANI", "DNIS", "SEGMENT", "QUEUE"], "objects" : { "Account" : [{ "displayName" : "Account Name", "apiName" : "Name" }] }, "screenPopSettings" : { "NoMatch" : { "screenPopType" : "PopToEntity", "screenPopData" : "Contact" }, "SingleMatch" : { "screenPopType" : "PopToVisualforce", "screenPopData" : "Visualforce_Page_Name" }, "MultipleMatches" : { "screenPopType" : "PopToSearch" } }, "Outbound" : { "callRelatedFields" : ["DNIS"], "objects" : { "Account" : [{ "displayName" : "Account Name", "apiName" : "Name" }] }, "screenPopSettings" : {} </pre>

名前	型	説明
		<pre> } } </pre>
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience の isSoftphonePanelVisible()

使用方法

ソフトフォンパネルの表示状況を返すには、このメソッドを使用します。ソフトフォンパネルが表示されている場合は `true`、最小化されている場合は `false` を返します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.isSoftphonePanelVisible({
  callback: function
});
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML とコールバック付きの JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };
    </script>
  </head>
</html>
```

```

        function isSoftphonePanelVisible() {
            sforce.opencti.isSoftphonePanelVisible({callback: callback});
        }
    </script>
</head>
<body>
    <button onclick="isSoftphonePanelVisible();">isSoftphonePanelVisible()</button>
</body>
</html>

```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	ソフトフォンパネルの表示状況を示す Boolean プロパティ <code>visible</code> が含まれます。ソフトフォンパネルが表示されている場合は <code>true</code> 、最小化されている場合は <code>false</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience の onClickToDial()

使用方法

有効化された電話番号をユーザがクリックするとコールされる関数を登録します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```

sforce.opencti.onClickToDial({
    listener: function
})

```

引数

名前	型	説明
listener	function	ユーザが有効化された電話番号をクリックしたときにコールされる JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var listener = function(payload) {
        console.log('Clicked phone number: ' + payload.returnValue.number);
      };

      // Register the listener.
      window.addEventListener('load', function() {
        sforce.opencti.onClickToDial({listener: listener});
      });
    </script>
  </head>
</html>
```

ペイロード

リスナメソッドへの各コールに渡されるペイロードオブジェクトには次の項目が含まれます。

名前	型	説明
number	number	ユーザがクリックした電話番号を提供します。
recordId	string	クリックされた電話番号に関連付けられたレコードの ID を提供します。
recordName	string	クリックされた電話番号のレコードの名前を提供します。
objectType	string	クリックされた電話番号に関連付けられたレコードのタイプを提供します。
accountId または contactId	string	クリックされた電話番号が個人取引先に属する場合、関連付けられた取引先または取引先責任者の ID が提供されます。
personAccount	boolean	クリックされた電話番号が個人取引先に属する場合、このプロパティは true です。 個人取引先が組織で有効化されていない場合、この項目はペイロードオブジェクトに含まれません。

Lightning Experience の onNavigationChange ()

使用方法

ページの URL が変わったか、ユーザが現在の場所から離れて移動したらコールする関数を登録します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.onNavigationChange({
  listener: function
});
```

引数

名前	型	説明
listener	function	ナビゲーションの変更時にコールされる JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var listener = function(payload) {
        console.log('Navigation change occurred. Payload: ', payload);
      };

      // Register the listener.
      window.addEventListener('load', function() {
        sforce.opencti.onNavigationChange({listener: listener});
      });
    </script>
  </head>
</html>
```

ペイロード

リスナメソッドへの各コールに渡されるペイロードオブジェクトには次の項目が含まれます。

名前	型	説明
url	string	ユーザの移動先ページの URL を提供します。

名前	型	説明
recordId	string	ユーザが取引先やケースなどの Salesforce レコードに移動した場合、読み込まれたレコードの ID が返されます。それ以外の場合、項目は空白になります。
recordName	string	ユーザが Salesforce レコードに移動した場合、読み込まれたレコードの名前。それ以外の場合、項目は空白になります。
objectType	string	ユーザが Salesforce レコードに移動した場合、読み込まれたオブジェクトの種別 (取引先やケースなど)。それ以外の場合、項目は空白になります。
accountId または contactId	string	ユーザの移動先ページが個人取引先のレコードホームである場合、関連付けられた取引先または取引先責任者の ID が返されます。
personAccount	boolean	ユーザの移動先ページが個人取引先のレコードホームである場合、この項目は <code>true</code> です。 組織で個人取引先が有効化されているが、ユーザの移動先ページが個人取引先のレコードホームではない場合、この項目は <code>false</code> です。 個人取引先が組織で有効化されていない場合、この項目はペイロードオブジェクトに含まれません。

Lightning Experience の refreshView()

使用方法

ビュー更新が呼び出された場合は `true`、それ以外の場合は `false` を返します。このメソッドが Salesforce コンソール内でコールされると、現在の有効なビューが更新されます。このタブに含まれている関連リストも更新されます。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.refreshView({
  callback: function
});
```

引数

名前	型	説明
callback	function	省略可能。API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML とコールバックなしの JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var param = {};
    function refreshView() {
      sforce.opencti.refreshView(param);
    }
  </script>
</head>
<body>
  <button onclick="refreshView();">refreshView</button>
</body>
</html>
```

サンプルコード – HTML とコールバック付きの JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var param = {};
    var callback = function(response) {
      if (response.success) {
        console.log('API method call executed successfully! returnValue:',
response.returnValue);
      } else {
        console.error('Something went wrong! Errors:', response.errors);
      }
    };
    param.callback = callback;

    function refreshView() {
      sforce.opencti.refreshView(param);
    }
  </script>
</head>
<body>
  <button onclick="refreshView();">refreshView</button>
```



```
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の runApex ()

使用方法

Salesforce で公開された Apex クラスから Apex メソッドを実行します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.runApex({
  apexClass:string,
  methodName:string,
  methodParams:string,
  callback:function //Optional
})
```

引数

名前	型	説明
args	object	JavaScript オブジェクトには以下が含まれます。 apexClass (文字列) — 実行するメソッドの Apex クラスを指定します。

名前	型	説明
		- methodName (文字列) — 実行するメソッドを指定します。 - methodParams (文字列) — メソッドに渡すパラメータを指定します。文字列は、項目値のペアを含み、有効なクエリ文字列の形式にする必要があります。 次に例を示します。 <pre>name=acme&phone=(212) 555-5555</pre> Apex メソッドがパラメータを取らない場合、methodParams を none または空のオブジェクト {} として指定できます。 - callback (関数) — メソッドの完了時にコールされる JavaScript メソッド。

サンプルコード – HTML および JavaScript

1. [設定] で、Apex クラスと Apex メソッドを作成します。

```
global class AccountRetrieval{

    webservice static String getAccount(String name) {
        List<Account> accounts = new List<Account>();
        for (Account account : Database.query('Select Id, Name, phone from Account where Name
like \'' + name + '%\')){
            accounts.add(account);
        }

        String jsonString = JSON.serialize(accounts);
        return jsonString;
    }
}
```

2. [設定] で、[WSDLからの生成]をクリックしてメソッドとクラスを公開し、サードパーティのソフトフォンからコールできるようにします。
3. コードをソフトフォンに追加します。

```
<html>
<head>
    <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
        var callback = function(response) {
            if (response.success) {
                console.log('API method call executed successfully! returnValue:',
response.returnValue);
            } else {
                console.error('Something went wrong! Errors:', response.errors);
            }
        };
    </script>
```

```

function runApex() {
    var param = {apexClass: 'AccountRetrieval', methodName: 'getAccount',
methodParams: 'name=acme'};
    param.callback = callback;
    //Invokes API method
    sforce.openct.runApex(param);
}
</script>
</head>
<body>
    <button onclick="runApex();">runApex</button>
</body>
</html>

```

4. 出力が返されます。ここでは、取引先 Acme が見つかりました。

```

{
  "success": true,
  "returnValue": {
    "runApex":
    "[{"attributes":{"type":"Account"},"url":"/services/data/v38.0/objects/Account/001xx000003DGawAAG"},"Id":"001xx000003DGawAAG"},"Name":"Acme"},"Phone":"(212) 555-5555"]"
  },
  "errors": null
}

```

応答

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は true、それ以外の場合は false が返されます。
returnValue	object	JavaScript オブジェクトには、指定された Apex クラスからメソッドを実行した結果が含まれます。返される形式の指定はできません。形式は、実行したメソッドの値によって決定されます。次に例を示します。

```

{"runApex":[{"attributes":{"type":"Account",
"url":"/services/data/v38.0/objects/Account/001xx000003DGawAAG"},"Id":"001xx000003DGawAAG",
"Name":"Acme"},"Phone":"(212) 555-5555"]}]}

```

名前	型	説明
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

関連トピック:

[Salesforce ヘルプ: Apex コードの概要](#)

Lightning Experience の saveLog()

使用方法

オブジェクトを Salesforce で保存または更新します。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.saveLog({
  value:{
    entityApiName:string, //Optional
    Id:string, //Optional
    param:value //Optional
  },
  callback:function //Optional
})
```

メモ:

- このメソッドを使用して更新するには、`Id` を含めます。
- このメソッドを使用して作成するには、`entityApiName` を含めます。
- 組織で個人取引先が有効になっている場合、このメソッドを使用して取引先または個人取引先を作成するには、取引先または個人取引先の `recordType ID` を含めます。`value` の引数には `recordType: recordTypeId` を含めます。次に例を示します。

```
sforce.opencti.saveLog({value: {entityApiName:"Account", recordType:"012R00000004U0y",
  LastName:'PersonAccountLast'}, callback:callback});
```

引数

名前	型	説明
value	object	保存または更新対象のオブジェクトの項目を指定します。 オブジェクトのIDが指定されると、レコードが更新されます。次に例を示します。 <pre>{Id:"00QR0000000yN5iMAE", LastName:"New lastname" }</pre> オブジェクトのIDが指定されない場合、新しいレコードが作成されます。次に例を示します。 <pre>{entityApiName:"Contact", LastName:"LastName" },callback:callback}</pre>  メモ: レコードを作成するには、すべての必須項目が指定されていることを確認します。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        console.log('API method call executed successfully! returnValue:',
response.returnValue);
      } else {
        console.error('Something went wrong! Errors:', response.errors);
      }
    }
    function saveLog() {
      //Update an existing object with the ID specified
      sforce.opencti.saveLog({value:{Id:"00QR0000000yN5iMAE", LastName:"New lastname"
}, callback:callback});
      //Create a contact
      sforce.opencti.saveLog({value:{entityApiName:"Contact", LastName:"LastName"
},callback:callback});
      //Update a lead
      sforce.opencti.saveLog({value:{Id:"00QR0000000yN5iMAE", LastName:"New lastname"
},callback:callback});
    }
  </script>
</head>
<body>
  <button onclick="saveLog();">saveLog</button>
```

```
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	オブジェクトが正常に作成または更新された場合はオブジェクトの ID、オブジェクトが正常に作成または更新されなかった場合は <code>null</code> 。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience の screenPop()

使用方法

入力種別およびパラメータの指定に従って新しい場所にポップします。このメソッドは API バージョン 38.0 以降で使用できます。

 **メモ:** OpenCTI for Lightning Experience では、値が「新規のブラウザウィンドウまたはタブ」の場合、ソフトウェアレイアウト項目「画面ポップの表示範囲」はサポートされません。Lightning Experience では、デフォルト値は「現在のブラウザウィンドウ」です。

構文

```
sforce.opencti.screenPop({
  type: sforce.opencti.SCREENPOP_TYPE.*, //Review the arguments section.
  params: object //Depends on the SCREENPOP_TYPE. Review the arguments section.
});
```

引数

名前	型	説明
type	string	画面をポップする種別の列挙。次のいずれかの値を使用します。 • <code>sforce.opencti.SCREENPOP_TYPE.SUBJECT</code> • <code>sforce.opencti.SCREENPOP_TYPE.URL</code> • <code>sforce.opencti.SCREENPOP_TYPE.OBJECTHOME</code> • <code>sforce.opencti.SCREENPOP_TYPE.LIST</code>
params	object	引数を保持するオブジェクト。種別によって異なります。 • SUBJECT の場合: <pre>params : { recordId: string }</pre> ここで、recordId は Salesforce の標準オブジェクトまたはカスタムオブジェクトの ID です。 • URL の場合: <pre>params : { url: string }</pre> URL は相対パラメータである必要があります。URL についての詳細は、『<i>Lightning コンポーネント開発者ガイド</i>』の <code>force:navigateToURL url</code> を参照してください。 • OBJECTHOME の場合: <pre>params : { scope: string }</pre> ケースや取引先などのオブジェクトまたはエンティティのホームにポップします。scope についての詳細は、『<i>Lightning コンポーネント開発者ガイド</i>』の <code>force:navigateToObject recordID</code> を参照してください。次にサンプル入力を示します。 <pre>{ scope: "Account" }</pre> • LIST の場合: <pre>{ listViewId: string, scope: string }</pre> listViewId および scope パラメータについての詳細は、『<i>Lightning コンポーネント開発者ガイド</i>』の <code>force:navigateToList</code> を参照してください。
callback	function	省略可能。API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML とコールバック付きの JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var callback = function(response) {
      if (response.success) {
        console.log('API method call executed successfully! returnValue:',
response.returnValue);
      } else {
        console.error('Something went wrong! Errors:', response.errors);
      }
    };
    function screenPop() {
      sforce.opencti.screenPop({type: sforce.opencti.SCREENPOP_TYPE.OBJECTHOME,
params: {scope:"Account"}, callback: callback });
    }
  </script>
</head>
<body>
  <button onclick="screenPop();">screenPop</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

関連トピック:

[Lightning コンポーネント開発者ガイド: force:navigateToURL](#)

[Lightning コンポーネント開発者ガイド: force:navigateToObject](#)

[Lightning コンポーネント開発者ガイド: force:navigateToList](#)

Lightning Experience の searchAndScreenPop()

使用方法

特定の文字列に対してソフトフォンレイアウトで指定されたオブジェクトを検索します。検索結果を返し、一致するレコードが画面にポップされます。このメソッドは、ソフトフォンレイアウトで定義された画面ポップ設定を優先します。このメソッドは API バージョン 38.0 以降で使用できます。

Lightning Experience の searchAndScreenPop() メソッドは、Salesforce Classic のメソッドと動作が異なります。

- Open CTI for Lightning Experience では、値が「新規のブラウザウィンドウまたはタブ」の場合、ソフトフォンレイアウト項目「画面ポップの表示範囲」はサポートされません。Lightning Experience では、デフォルト値は「現在のブラウザウィンドウ」です。
- Open CTI for Lightning Experience には、deferred という新しい引数があります。

 **ヒント:** searchAndGetScreenPopUrl() メソッドは、Open CTI API for Lightning Experience では使用できません。Lightning で同じ機能を実現するには、このメソッドで利用できる deferred パラメータを使用します。返されたオブジェクトから screenPop() メソッドに SCREEN_POP_DATA 内の値を渡します。

ソフトフォンレイアウトのデフォルト設定で動作に矛盾がある場合には、ソフトフォンレイアウトを編集してキャッシュを強制的に更新します。[設定]からソフトフォンレイアウトを編集し、変更を保存します。次に、レイアウトをもう一度編集し、レイアウトをデフォルト設定にリセットします。

構文

```
sforce.opencti.searchAndScreenPop({
  searchParams:string //Optional
  queryParams:string, //Optional
  callType:sforce.opencti.CALL_TYPE.*, //Required. See arguments for more information.

  deferred:boolean //Optional
  callback:function //Optional
});
```

引数

名前	型	説明
searchParams	string	検索対象の文字列。
queryParams	string	URL に渡すクエリパラメータを指定します。画面ポップオプションが Pop to Visualforce に設定されている場合のみ、クエリパラメータが URL に渡されます。

名前	型	説明
callType	string	着信、発信、内線、または null などの通話種別を指定します。通話種別は、ソフトフォンレイアウトでの設定ごとに、語句検索を行うオブジェクトを決定します。 次のいずれかの値によって通話種別を指定します。 • sforce.opencti.CALL_TYPE.INBOUND • sforce.opencti.CALL_TYPE.OUTBOUND • sforce.opencti.CALL_TYPE.INTERNAL
deferred	boolean	画面ポップが検索の直後に実行されるか後で実行されるかを指定します。画面ポップが後で実行される場合は、JSON オブジェクトが返されます。操作を実行するには、このオブジェクトを変更せずに sforce.opencti.screenPop に渡す必要があります。 • False - デフォルト値。検索の実行直後に画面ポップが実行されることを示します。 • True - SCREEN_POP_DATA キー内で JSON {object} が返されます。画面ポップを実行するには、このオブジェクトを変更せずに sforce.opencti.screenPop に返します。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var callback = function(response) {
      if (response.success) {
        console.log('API method call executed successfully! returnValue:',
response.returnValue);
      } else {
        console.error('Something went wrong! Errors:', response.errors);
      }
    };
    function searchAndScreenPop() {
      //Invokes API method
      sforce.opencti.searchAndScreenPop({ searchParams : 'Acme',queryParams :
'Key1=value1&Key2=value2', callType : sforce.opencti.CALL_TYPE.INBOUND, deferred: false,
callback : callback });
    }
  </script>
</head>
<body>
  <button onclick="searchAndScreenPop();">searchAndScreenPop</button>
```

```
</body>
</html>
```

応答

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	検索結果に一致するオブジェクトのリストを返します。検索は、ソフトフォンレイアウトで指定されたオブジェクトで実行されます。検出されたオブジェクトごとに、オブジェクト ID、オブジェクトタブ名、項目名、および項目値が JSON オブジェクトとして返されます。

次に、「Acme」を検索し、1つの取引先オブジェクトと3つの商談オブジェクトが検出された例を示します。

```
{
  "006x0000001ZcyG":{
    "Name":"Acme - 600 Widgets",
    "object":"Opportunity",
    "displayName":"Opportunity"
  },
  "001x0000003DGQR":{
    "Name":"Acme",
    "Type":"Analyst",
    "object":"Account",
    "displayName":"Company"
  },
  "006x0000001ZcyH":{
    "Name":"Acme - 200 Widgets",
    "object":"Opportunity",
    "displayName":"Opportunity"
  },
  "006x0000001ZcyF":{
    "Name":"Acme - 1,200 Widgets",
    "object":"Opportunity",
    "displayName":"Opportunity"
  }
}
```

この API メソッドを `deferred` パラメータを指定して呼び出すと、画面は移動せずに次のサンプル出力が返されます。

```
{
  "006x0000001ZcyG":{
    "Name":"Acme - 600 Widgets",
    "object":"Opportunity",
    "displayName":"Opportunity"
  },
}
```

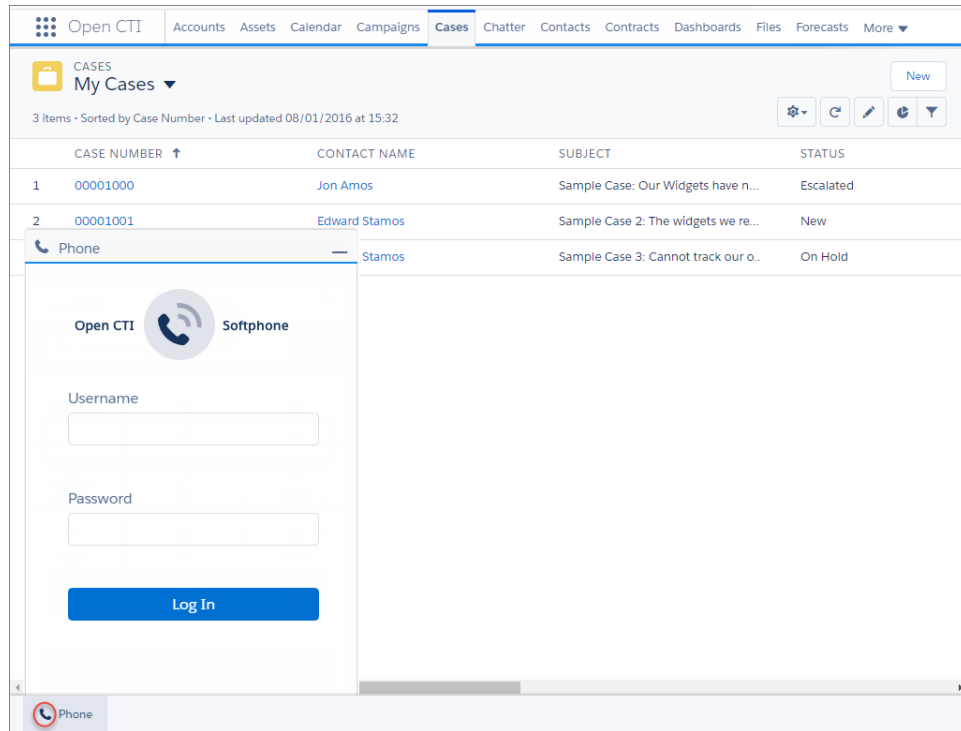
名前	型	説明
		<pre>SCREEN_POP_DATA : {} //an object. Do not modify it. Pass it to screenPop() API to navigate. }</pre> 個人取引先でこの API メソッドを呼び出すと、追加情報が返されます。 • <code>accountId</code> または <code>contactId</code> — 関連する取引先または取引先責任者の ID。 • <code>personAccount</code> - 組織で個人取引先が有効になっている場合は <code>true</code>、それ以外の場合は <code>false</code>。 次に例を示します。 <pre>{ "006x0000001ZcyG":{ "RecordName":"Acme Person Account", "RecordType":"Account", "PersonContactId":"003D000000QOMqg", "IsPersonAccount":true } } { "url":"http://<i>yourInstance</i>.salesforce.com/001x0000003DGQR", "RecordId":"001x0000003DGQR", "RecordName":"Acme Person Account", "RecordType":"Account", "PersonContactId":"003D000000QOMqg", "IsPersonAccount":true }</pre>
<code>errors</code>	<code>array</code>	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience の setSoftphoneItemIcon()

使用方法

ユーティリティバーのソフトフォン項目のアイコンを設定します。関数の実行が成功した場合は `true` を返し、失敗した場合は `false` を返します。このメソッドは API バージョン 38.0 以降で使用できます。

ユーティリティバーのソフトフォンアイコン。



構文

```
sforce.opencti.setSoftphoneItemIcon({
  key:key,
  callback:function //Optional
});
```

引数

名前	型	説明
key	string	ユーティリティーバーのソフトフォンアイコンに使用する Lightning Design System 内のアイコンに対応するキー。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var callback = function(response) {
```

```

        if (response.success) {
            console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
            console.error('Something went wrong! Errors:', response.errors);
        }
    };

    function setSoftphoneItemIcon() {
        sforce.opencti.setSoftphoneItemIcon({key:"call", callback: callback});
    }
</script>
</head>
<body>
    <button onclick="setSoftphoneItemIcon();">setSoftphoneItemIcon()</button>
</body>
</html>

```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールの呼び出しが成功した場合は <code>true</code> 、それ以外の場合は <code>false</code> 。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> です。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

関連トピック:

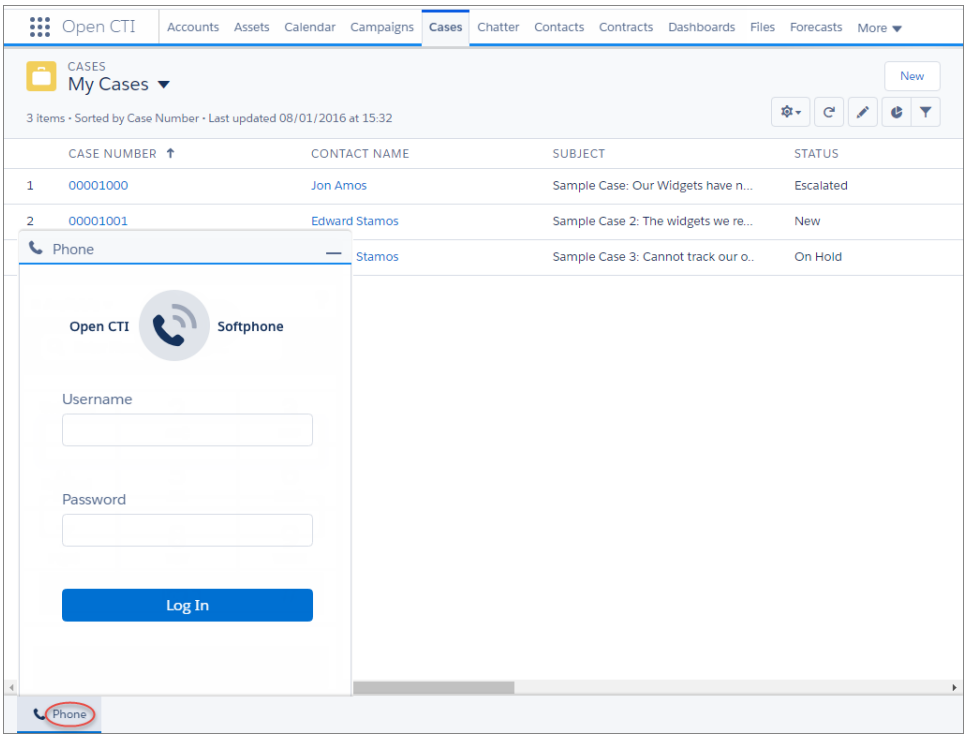
[Salesforce Lightning Design System: ユーティリティアイコン](#)

Lightning Experience の setSoftphoneItemLabel()

使用方法

ユーティリティバーのソフトフォンコンポーネント項目の表示ラベルを設定します。関数の実行が成功した場合は `true` を返し、失敗した場合は `false` を返します。このメソッドは API バージョン 38.0 以降で使用できます。

ユーティリティバーのソフトフォン表示ラベル。



構文

```
sforce.opencti.setSoftphoneItemLabel({
  label: string,
  callback:function //Optional
});
```

引数

名前	型	説明
label	string	ユーティリティバーのソフトフォン表示ラベルに使用する文字列。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
```

```

        if (response.success) {
            console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
            console.error('Something went wrong! Errors:', response.errors);
        }
    };

    function setSoftphoneItemLabel() {
        sforce.opencti.setSoftphoneItemLabel({label: "MySoftphone", callback: callback});
    }
</script>
</head>
<body>
    <button onclick="setSoftphoneItemLabel();">setSoftphoneItemLabel()</button>
</body>
</html>

```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールの呼び出しが成功した場合は <code>true</code> 、それ以外の場合は <code>false</code> 。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> です。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience の setSoftphonePanelHeight()

使用方法

ユーティリティバーのソフトフォンパネルの高さを設定します。高さはピクセル単位で指定する必要があります。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```

sforce.opencti.setSoftphonePanelHeight({
    heightPX:height,

```



```
    callback:function //Optional
  });
```

引数

名前	型	説明
heightPX	number	ソフトフォンの高さ(ピクセル単位)。高さは 240 ～ 700 の数値である必要があります。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
  <script type="text/javascript">
    var callback = function(response) {
      if (response.success) {
        console.log('API method call executed successfully! returnValue:',
response.returnValue);
      } else {
        console.error('Something went wrong! Errors:', response.errors);
      }
    };

    function setSoftphonePanelHeight() {
      sforce.opencti.setSoftphonePanelHeight({heightPX: 400, callback: callback});
    }
  </script>
</head>
<body>
  <button onclick="setSoftphonePanelHeight();">setSoftphonePanelHeight()</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。

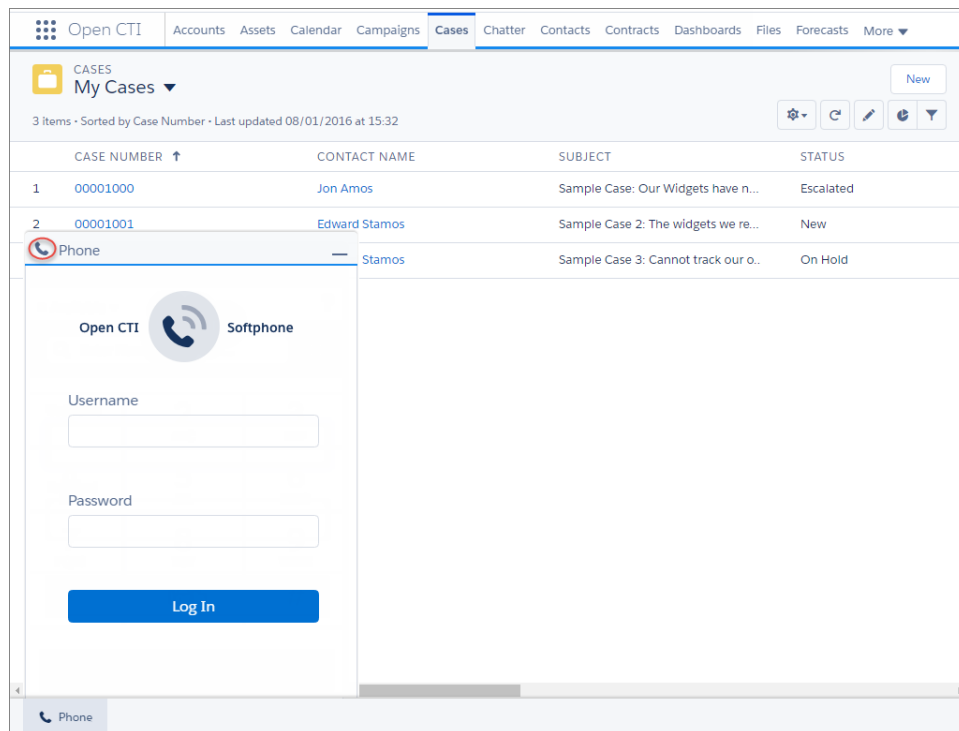
名前	型	説明
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に null になります。
errors	array	API コールが正常に行われた場合、この変数は null になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の setSoftphonePanelIcon()

使用方法

ソフトフォンパネルのアイコンを設定します。関数の実行が成功した場合は true を返し、失敗した場合は false を返します。このメソッドは API バージョン 38.0 以降で使用できます。

ソフトフォンパネルのアイコン。



構文

```
sforce.opencti.setSoftphonePanelIcon({
  key: key,
```

```
    callback:function //Optional
  });
```

引数

名前	型	説明
key	string	ソフトフォンパネルのアイコンに使用する Lightning Design System 内のアイコンに対応するキー。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function setSoftphonePanelIcon() {
        sforce.opencti.setSoftphonePanelIcon({key:"call", callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="setSoftphonePanelIcon();">setSoftphonePanelIcon()</button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールの呼び出しが成功した場合は true、それ以外の場合は false。

名前	型	説明
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に null になります。
errors	array	API コールが正常に行われた場合、この変数は null です。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

関連トピック:

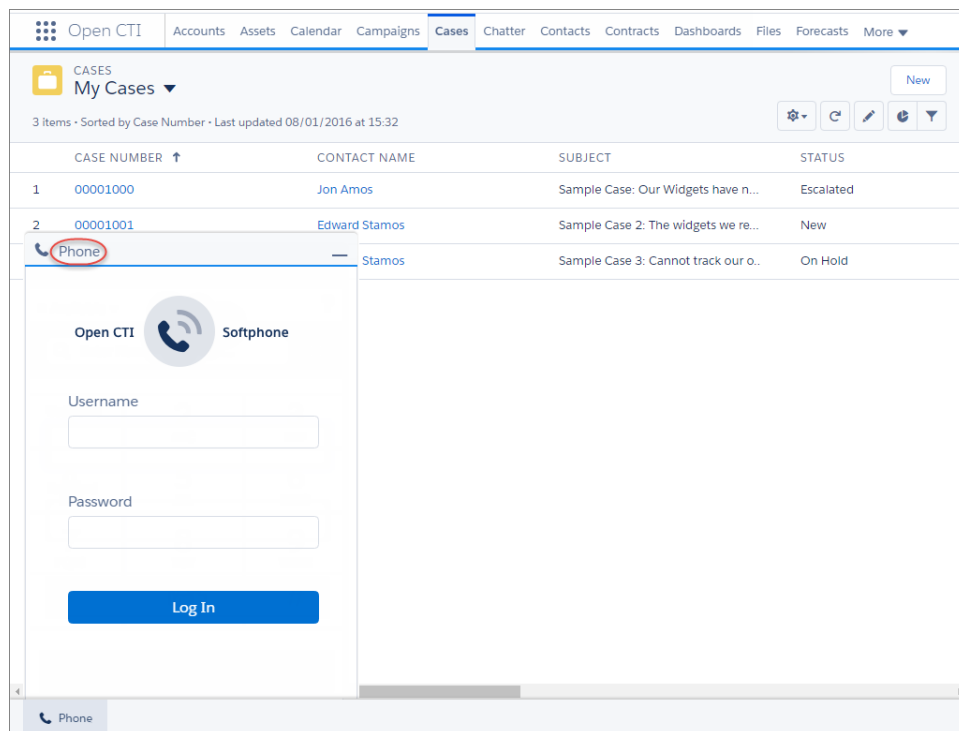
[Salesforce Lightning Design System: ユーティリティアイコン](#)

Lightning Experience の setSoftphonePanelLabel ()

使用方法

ソフトフォンパネルの表示ラベルを設定します。関数の実行が成功した場合は true を返し、失敗した場合は false を返します。このメソッドは API バージョン 38.0 以降で使用できます。

ソフトフォンパネルの表示ラベル。



構文

```
sforce.opencti.setSoftphonePanelLabel({
  label: string,
  callback: function //Optional
});
```

引数

名前	型	説明
label	string	ソフトフォンパネルの表示ラベルに使用する文字列。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function setSoftphonePanelLabel() {
        sforce.opencti.setSoftphonePanelLabel({label: "Mysoftphone",callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="setSoftphonePanelLabel();">setSoftphonePanelLabel()</button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールの呼び出しが成功した場合は true、それ以外の場合は false。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に null になります。
errors	array	API コールが正常に行われた場合、この変数は null です。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の setSoftphonePanelVisibility()

使用方法

ソフトフォンパネルの表示状況を設定します。visible パラメータが true として渡されると、ソフトフォンパネルが表示されます。false に設定されていると、パネルは最小化されます。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.setSoftphonePanelVisibility({
  visible: true,
  callback: function //Optional
});
```

引数

名前	型	説明
visible	boolean	ソフトフォンパネルをドッキング (表示) するには、値を true に設定します。ソフトフォンパネルを最小化 (非表示) するには、値を false に設定します。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
```

```

    var callback = function(response) {
        if (response.success) {
            console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
            console.error('Something went wrong! Errors:', response.errors);
        }
    };

    function setSoftphonePanelVisibility() {
        sforce.opencti.setSoftphonePanelVisibility({visible: true, callback: callback});
    }
</script>
</head>
<body>
    <button onclick="setSoftphonePanelVisibility();">setSoftphonePanelVisibility()</button>

</body>
</html>

```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は true、それ以外の場合は false が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に null になります。
errors	array	API コールが正常に行われた場合、この変数は null になります。API コールが失敗した場合、この変数はエラーメッセージの配列を返します。

Lightning Experience の setSoftphonePanelWidth()

使用方法

ユーティリティバーのソフトフォンパネルの幅を設定します。幅はピクセル単位で指定する必要があります。このメソッドは API バージョン 38.0 以降で使用できます。

構文

```
sforce.opencti.setSoftphonePanelWidth({
  widthPX:width,
  callback:function //Optional
});
```

引数

名前	型	説明
widthPX	number	ソフトフォンの幅 (ピクセル単位)。高さは 200 ～ 1240 の数値である必要があります。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – HTML および JavaScript

```
<html>
  <head>
    <script type="text/javascript"
src="https://domain:port/support/api/38.0/lightning/opencti_min.js"></script>
    <script type="text/javascript">
      var callback = function(response) {
        if (response.success) {
          console.log('API method call executed successfully! returnValue:',
response.returnValue);
        } else {
          console.error('Something went wrong! Errors:', response.errors);
        }
      };

      function setSoftphonePanelWidth() {
        sforce.opencti.setSoftphonePanelWidth({widthPX: 400, callback: callback});
      }
    </script>
  </head>
  <body>
    <button onclick="setSoftphonePanelWidth();">setSoftphonePanelWidth()</button>
  </body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
success	boolean	API メソッドコールが正常に呼び出された場合は <code>true</code> 、それ以外の場合は <code>false</code> が返されます。
returnValue	object	この API メソッドはこのオブジェクトを返しません。returnValue は常に <code>null</code> になります。
errors	array	API コールが正常に行われた場合、この変数は <code>null</code> になります。API コールが失敗した場合、この変数は エラーメッセージ の配列を返します。

Lightning Experience メソッドの一般的なエラーメッセージ

すべての Lightning Experience メソッドで、エラーオブジェクトは配列として返されます。

エラーオブジェクトの一部として次の項目が含まれます。

code: *string*

エラーコードを示す定数文字列。

description: *string*

エラーコードの説明。

details: *object*

通常は定義されません。この定数には、saveLog メソッドのエラーオブジェクトに関する詳細を含めることができます。

サンプルエラーオブジェクト

```
[{
  code: code1
  description: description1
  details: details1
},{
  code: code2
  description: description2
  details: details2
}]
```

INVALID_PARAM エラーコードのサンプルエラーオブジェクト

```
[{
  code: "INVALID_PARAM",
  description: "An invalid value was passed to the parameter parameterName. A numeric value was expected, but undefined was found instead."
}]
```

GENERIC_PARAM エラーコードのサンプルエラーオブジェクト

```
[{
  code: "GENERIC_ERROR",
```

```

    description: "An error occurred while calling the API method."
  }]

```

SERVER_ERROR コードのサンプルエラーオブジェクト

```

[ {
  code: "SERVER_ERROR",
  description: "A problem was encountered on the server."
}]

```

SOFTPHONE_CONTAINER_ERROR コードのサンプルエラーオブジェクト

```

[ {
  code: "SOFTPHONE_CONTAINER_ERROR",
  description: "Unable to execute sendPostMessage because the softphone container hasn't initialized yet."
}]

```

runApex メソッドでサーバエラーがある場合、description 項目には "Could not load Apex class: apexClassName." が含まれます。

saveLog メソッドの場合、details 項目にはエラーの種類に基づく情報が含まれます。次に例を示します。

```

[ {
  code: "GENERIC_ERROR",
  description: "An error occurred while calling the saveLog() API method. Review the Details field in the error section.",
  details: [ {
    message: "An error occurred while trying to update the record. Please try again.",

    pageErrors: [],
    fieldErrors: {
      Name: [ {
        statusCode: "REQUIRED_FIELD_MISSING",
        message: "Required fields are missing: [Name]",
        fieldLabel: "Account Name",
        columnApiName: "Name"
      } ]
    },
    potentialDuplicates: []
  } ]
}]

```

第 5 章 Salesforce Classic のメソッド

組織で Salesforce Classic を使用している場合、Salesforce Classic で機能するメソッドを使用します。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

Salesforce アプリケーション操作のメソッド

Open CTI を使用すると、ケースフィードページの要素を含む、Salesforce アプリケーションを CTI システムから操作できます。

コンピュータテレフォニーインテグレーション (CTI) のメソッド

Open CTI を使用すると、CTI システムを Salesforce に統合できます。

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

[Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性](#)

Salesforce アプリケーション操作のメソッド

Open CTI を使用すると、ケースフィードページの要素を含む、Salesforce アプリケーションを CTI システムから操作できます。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

[Salesforce Classic の getPageInfo\(\)](#)

[Salesforce Classic の isInConsole\(\)](#)

[Salesforce Classic の isVisible\(\)](#)

[Salesforce Classic の notifyInitializationComplete\(\)](#)

[Salesforce Classic の onFocus\(\)](#)

[Salesforce Classic の onObjectUpdate\(\)](#)

[Salesforce Classic の refreshObject\(\)](#)
[Salesforce Classic の refreshPage\(\)](#)
[Salesforce Classic の refreshRelatedList\(\)](#)
[Salesforce Classic の reloadFrame\(\)](#)
[Salesforce Classic の runApex\(\)](#)
[Salesforce Classic の saveLog\(\)](#)
[Salesforce Classic の screenPop\(\)](#)
[Salesforce Classic の searchAndGetScreenPopUrl\(\)](#)
[Salesforce Classic の searchAndScreenPop\(\)](#)
[Salesforce Classic の setVisible\(\)](#)

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)
[Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性](#)

Salesforce Classic の getPageInfo()

使用方法

現在のページに関する情報を返します。

構文

```
sforce.interaction.getPageInfo(callback: function);
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```

<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      }
    };
  </script>
</head>
</html>

```

```

        } else {
            alert(response.error);
        }
    };
    function getPageInfo() {
        //Invokes API method
        sforce.interaction.getPageInfo(callback);
    }
</script>
</head>
<body>
    <button onclick="getPageInfo();">getPageInfo</button>
</body>
</html>

```

応答

名前	型	説明
result	string	現在のページの URL を JSON 文字列として返します。該当するオブジェクト ID、オブジェクト名、オブジェクト種別、および API バージョン 33.0 以降の場合はオブジェクトタブ名が含まれます。次に例を示します。 <pre>{ "url": "http://na1.salesforce.com/001x0000003DGQR", "objectId": "001x0000003DGQR", "objectName": "Acme", "object": "Account", "displayName": "Company" }</pre> API バージョン 31.0 以降では、個人取引先オブジェクトで API メソッドを呼び出すと次の追加情報を返します。 - AccountId または ContactId、関連する取引先または取引先責任者の ID - PersonAccount、オブジェクトが個人取引先の場合は true、それ以外の場合は false 次に例を示します。 <pre>{ "url": "http://na1.salesforce.com/001x0000003DGQR", "objectId": "001x0000003DGQR", "objectName": "Acme Person Account", "object": "Account", "contactId": "003D000000QOMqg", "personAccount": true }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の `isInConsole()`

使用方法

ソフトフォンが Salesforce コンソールにあるかどうかを示します。

 **メモ:** このメソッドはマルチモニターコンポーネントが有効になっている Salesforce コンソールで 사용되는場合、ポップアウトしたソフトフォンコンポーネントがコンソール内と同様に示されます。

構文

```
sforce.interaction.isInConsole(callback: function)
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('User is in console.');      }
      else {
        alert('User is not in console.');      }
    };
  </script>
</head>
<body>
  <button onclick="sforce.interaction.isInConsole(callback);">isInConsole</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	boolean	ソフトフォンが Salesforce コンソールにある場合は true、ソフトフォンが Salesforce コンソールにない場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: Salesforce コンソールのマルチモニターコンポーネントの有効化](#)

Salesforce Classic の isVisible()

使用方法

ソフトフォンが表示される場合は true、非表示の場合は false を返します。

構文

```
sforce.interaction.isVisible(callback: function)
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Softphone is visible');
      } else {
        alert('Softphone is not visible');
      }
    };
    function isVisible() {
      sforce.interaction.isVisible(callback);
    }
  </script>
</head>
</html>
```

```

</script>
</head>
<body>
    <button onclick="isVisible();">isVisible</button>
</body>
</html>

```

応答

名前	型	説明
result	boolean	ソフトフォンが表示される場合は true、非表示の場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の notifyInitializationComplete()

使用方法

Salesforce に、ソフトフォンの初期化が完了したことで、Salesforce がスタンバイ URL に切り替えないようにすることを通知します。ソフトフォンの初期化中は、[ソフトフォン] 領域に読み込み中アイコンが表示されます。スタンバイ URL を使用するには、コールセンターの定義ファイルに指定する必要があります。詳細は、「[コールセンターの省略可能な要素と属性](#)」(ページ 14)を参照してください。

構文

```
sforce.interaction.cti.notifyInitializationComplete()
```

引数

なし。

サンプルコード

```

<html>
<head>
<script src="http://domain:port/support/api/29.0/interaction.js"></script>
<script type="text/javascript">
    // Invokes API method
    sforce.interaction.cti.notifyInitializationComplete();
</script>
</head>
<body>
    The interaction framework has been notified that the CTI initialization is complete.

```



```
</body>
</html>
```

応答

なし。

Salesforce Classic の **onFocus()**

使用方法

ブラウザのフォーカスが変更されるとコールされる関数を登録します。Salesforce コンソールでは、主タブまたはナビゲーションタブでユーザが移動すると、ブラウザのフォーカスが変更されます。

構文

```
sforce.interaction.onFocus( listener:function );
```

引数

名前	型	説明
listener	function	ブラウザのフォーカスに変更されたときにコールされる JavaScript メソッド。

サンプルコード – JavaScript

```
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      }
    };
    function onFocus() {
      //Invokes API method
      sforce.interaction.onFocus(callback);
    }
  </script>
</head>
<body>
  <button onclick="onFocus();">onFocus</button>
</body>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	string	フォーカスされているページの URL を JSON 文字列として返します。該当するオブジェクト ID、オブジェクト名、オブジェクト種別、および API バージョン 33.0 以降の場合はオブジェクトタブ名が含まれます。以下に例を示します。 <pre>{ "url": "http://salesforce.com/001x0000003DGQR", "objectId": "001x0000003DGQR", "objectName": "Acme", "object": "Account", "displayName": "Company" }</pre> ページがオブジェクトにフォーカスしていない場合、オブジェクト ID、オブジェクト名、およびオブジェクトは空になります。 API バージョン 31.0 以降では、個人取引先オブジェクトで API メソッドを呼び出すと次の追加情報を返します。 - AccountId または ContactId、関連する取引先または取引先責任者の ID - PersonAccount、オブジェクトが個人取引先の場合は true、それ以外の場合は false 次に例を示します。 <pre>{ "url": "http://na1.salesforce.com/001x0000003DGQR", "objectId": "001x0000003DGQR", "objectName": "Acme Person Account", "object": "Account", "contactId": "003D000000QOMqg", "personAccount": true }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: Salesforce コンソールのマルチモニターコンポーネントの有効化](#)

Salesforce Classic の onObjectUpdate()

使用方法

ケース項目、フィールド、または関連リストデータが、フィールドベースのレイアウトで表示されているレコードで変更されるとコールされる関数を登録します。

 **メモ:** [ケースフィード]のカスタムパブリッシャーとして使用する Visualforce ページでこのメソッドを使用します。

構文

```
sforce.interaction.entityFeed.onObjectUpdate(callback: function)
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<apex:page standardController="Case">
  <apex:includeScript value="/support/api/26.0/interaction.js"/>
  <script type="text/javascript">
    var callback = function(response) {
      alert('Case was updated. Fields = ' + response.fieldsUpdated +
        ' Related lists = ' + response.relatedListsUpdated + ' Feed = ' +
          response.feedUpdated);
    };
    //Invokes API method
    sforce.interaction.entityFeed.onObjectUpdate(callback);
  </script>
</apex:page>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
fieldsUpdated	boolean	1 つ以上のケース項目が更新された場合は true。
relatedListsUpdated	boolean	1 つ以上のケース関連リストが更新されなかった場合は true。
feedUpdated	boolean	ケースフィードが更新された場合は true。

Salesforce Classic の refreshObject ()

使用方法

フィードベースのレイアウトを使用するページに、その項目、フィード、または関連リストデータが変更されたことを通知し、これらをページで強制的に更新します。

 **メモ:** [ケースフィード]のカスタムパブリッシャーとして使用する Visualforce ページでこのメソッドを使用します。

構文

```
sforce.interaction.entityFeed.refreshObject (
  objectId:string,
  refreshFields:boolean,
  refreshRelatedLists:boolean,
  refreshFeed:boolean, callback:function)
```

引数

名前	型	説明
objectId	string	ケースオブジェクトのレコード ID。
refreshFields	boolean	ケースの 1 つ以上の項目が変更されたことを示します。
refreshRelatedLists	boolean	1 つ以上のケース関連リストが変更されたことを示します。
refreshFeed	boolean	ケースフィードが変更されたことを示します。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<apex:page standardController="Case">
  <apex:includeScript value="/support/api/26.0/interaction.js"/>
  <a href="javascript:void(0);"
  onclick="sforce.interaction.entityFeed.refreshObject('{!case.id}', true, true, true,
  function(response) {alert('Case was updated: ' + response.result);});">Refresh Case</a>
</apex:page>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	boolean	[ケースフィード]ページが正常に更新された場合は true、それ以外の場合は false。

Salesforce Classic の refreshPage ()

使用方法

ページ更新が呼び出された場合は true、それ以外の場合は false を返します。このメソッドが Salesforce コンソール内でコールされると、現在の有効なタブが更新されます。このメソッドは、API バージョン 28.0 以降でのみ使用できます。

構文

```
sforce.interaction.refreshPage (callback: function);
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/28.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Page refresh has been invoked.');
      } else {
        alert('Page refresh has not been invoked.');
      }
    };
    function refreshPage() {
      sforce.interaction.refreshPage (callback);
    }
  </script>
</head>
<body>
  <button onclick="refreshPage();">refreshPage</button>
</body>
</html>
```

応答

名前	型	説明
result	boolean	ページ更新が呼び出された場合は true、それ以外の場合は false を返します。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の refreshRelatedList()

使用方法

指定された listName を持つ関連リストが更新された場合は true、それ以外の場合は false を返します。このメソッドが Salesforce コンソール内でコールされると、現在フォーカスのあるビューで指定のリスト名を持つ関連リストのみが更新されます。このメソッドは、API バージョン 28.0 以降でのみ使用できます。

構文

```
sforce.interaction.refreshRelatedList(listName:string, callback:function)
```

引数

名前	型	説明
listName	string	更新する関連リストの名前。たとえば、取引先責任者関連リストの取引先責任者や活動予定関連リストの活動。 カスタム参照項目から作成されたカスタム関連リストを更新するには、listName は、カスタム参照項目の ID を指定する必要があります。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/28.0/interaction.js"></script>
  <script type="text/javascript">
    function checkRefreshResult(result) {
      if (result.result) {
        alert('The related list is refreshed!');
      }
    }
  </script>
</head>
<body>
  <div>
    <button type="button" value="Refresh">Refresh</button>
  </div>
</body>
</html>
```

```

        } else {
            alert('Cannot refresh the related list with the given listName! Make
sure the listName is correct and the related list is on the page.');
```

```

        }
    }
    function refreshActivityRelatedList() {
        sforce.interaction.refreshRelatedList('Activity', checkRefreshResult);
    }

    function refreshHistoryRelatedList() {
        sforce.interaction.refreshRelatedList('History', checkRefreshResult);
    }

    function saveAndRefresh() {
        sforce.interaction.saveLog('Task',
'Subject=ImportantTask&WhatId=[15-character ID of an account to which you want to attach
the task]', function(result) {
            if (result.result) {
                refreshActivityRelatedList();
            } else {
                alert('Could not save the object! Check the developer console for error
messages.');
```

```

            }
        });
    }
</script>
</head>
<body>
    <button onclick="refreshHistoryRelatedList();">Refresh History Related List</button>

    <button onclick="saveAndRefresh();">Save and Refresh</button>
</body>
</html>

```

応答

名前	型	説明
result	boolean	指定された名前を持つ関連リストが更新された場合は true、それ以外の場合は false を返します。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

メモ

- このメソッドは、<apex:relatedList> から作成された関連リストを更新できません。
- このメソッドは、Salesforce コンソールで上書きされた Visualforce ページから関連リストを更新できません。
- Salesforce コンソール内からコールされると、このメソッドは現在フォーカスされているビューの更新対象の関連リストの検索のみを行います。

Salesforce Classic の reloadFrame ()

使用方法

コールを実行するページを含むフレームを再読み込みします。このメソッドは、レコードがフィードベースのレイアウトで表示されている場合にのみ使用できます。このメソッドは API バージョン 34.0 以降で使用できます。

構文

```
sforce.interaction.entityFeed.reloadFrame()
```

引数

なし。

サンプルコード – JavaScript

```
<apex:page standardController="Case">
  <apex:includeScript value="/support/api/34.0/interaction.js"/>
  <a href="javascript:void(0); onclick="sforce.interaction.entityFeed.reloadFrame();">
    Reload</a>
</apex:page>
```

応答

なし。

Salesforce Classic の runApex ()

使用方法

Salesforce で公開された Apex クラスから Apex メソッドを実行します。

構文

```
sforce.interaction.runApex(apexClass:string, methodName:string, methodParams:string,
(optional) callback:function)
```


引数

名前	型	説明
apexClass	string	実行するメソッドの Apex クラスを指定します。
methodName	string	実行するメソッドを指定します。
methodParams	string	メソッドに渡すパラメータを指定します。文字列は、項目値のペアを含み、有効なクエリ文字列の形式にする必要があります。たとえば、name=acme&phone=(212) 555-5555 です。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード - JavaScript

1. システム管理者が Apex クラスと Apex メソッドを作成します。

```
global class AccountRetrieval{

WebService static String getAccount(String name) {
    List<Account> accounts = new List<Account>();
    for (Account account : Database.query('Select Id, Name, phone from Account where Name
like \'' + name + '%\')){
        accounts.add(account);
    }

    String jsonString = JSON.serialize(accounts);
    return jsonString;
}
}
```

2. サードパーティのソフトフォンがコールできるように、Salesforce の Apex クラスおよび Apex メソッドを作成した場所で [WSDL の作成] をクリックしてメソッドおよびクラスを公開します。
3. コードをソフトフォンに追加します。

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      } else {
        alert(response.error);
      }
    };
    function runApex() {
      //Invokes API method
      sforce.interaction.runApex('AccountRetrieval', 'getAccount', 'name=acme',
callback);
    }
  </script>
</head>
</html>
```

```

</script>
</head>
<body>
    <button onclick="runApex();">runApex</button>
</body>
</html>

```

4. 出力が返されます。ここでは、取引先 Acme が見つかりました。

```

[{"attributes":{"type":"Account",
"url":"/services/data/v25.0/objects/Account/001x0000003DGQRAA4"},
"Id":"001x0000003DGQRAA4","Name":"Acme","Phone":"(212) 555-5555"}]

```

応答

名前	型	説明
result	string	指定された Apex クラスからメソッドを実行した結果を返します。 返される形式の指定はできません。形式は、実行されたメソッドの値によって決定されます。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Apex コードの概要](#)

Salesforce Classic の saveLog ()

使用方法

オブジェクトを Salesforce で保存または更新します。

構文

```
sforce.interaction.saveLog(object:string, saveParams:string, (optional)callback:function)
```

引数

名前	型	説明
object	string	保存または更新対象のオブジェクトの名前。

名前	型	説明
saveParams	string	保存または更新対象のオブジェクトの項目を指定します。 オブジェクトのIDが指定されると、レコードが更新されます。たとえば、Id=001D000000J6qIX&Name=Acme&Phone=4154561515 です。オブジェクトのIDが指定されない場合、新しいレコードが作成されます。たとえば、Name=Acme&Phone=4154561515 です。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      } else {
        alert(response.error);
      }
    }
    function saveLog() {
      //Invokes API method
      sforce.interaction.saveLog('Account', 'Name=NewAccountName&Phone=4155551212',
callback);
    }
  </script>
</head>
  <button onclick="saveLog();">saveLog</button>
</html>
```

応答

名前	型	説明
result	boolean	オブジェクトが正常に保存または更新された場合は true、オブジェクトが正常に保存または更新されなかった場合は false。
id	string	新たに作成されたオブジェクトの ID。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の screenPop()

使用方法

対象 URL (相対である必要がある) にポップします。

構文

```
sforce.interaction.screenPop(url:string, force:boolean, (optional) callback:function)
```

引数

名前	型	説明
url	string	相対 URL。画面ポップの場所を指定します。
force	boolean	画面ポップを強制的に表示するには値を true に設定し、それ以外の場合は false に設定します。この引数は、API バージョン 28.0 以降でのみ使用できます。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/28.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Screen pop was set successfully.');
      }
      else {
        alert('Screen pop failed.' + result.error);
      }
    };
    function screenPop() {
      //Invokes API method
      sforce.interaction.screenPop('/001x0000003DGQR', true, callback);
    }
  </script>
</head>
<body>
  <!-- Note that '001x0000003DGQR' is an example of an object Id to screen pop. -->
  <button onclick="screenPop();">screen pop to entity Id</button>
</body>
</html>
```

応答

名前	型	説明
result	boolean	画面ポップが正常に表示された場合は true、正常に表示されなかった場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の searchAndGetScreenPopUrl ()

使用方法

特定の文字列に対してソフトフォンレイアウトで指定されたオブジェクトを検索します。検索結果と、ポップ画面への相対URLを返します。このメソッドは、実際の画面ポップを実行しません。このメソッドは、ソフトフォンレイアウトで定義された画面ポップ設定を優先します。このメソッドは、API バージョン 28.0 以降でのみ使用できます。

 **ヒント:** このメソッドは、Open CTI API for Lightning Experience では使用できません。Lightning で同じ機能を実現するには、[Lightning Experience の searchAndScreenPop \(\)](#) メソッドで使用する deferred パラメータを使用します。

構文

```
sforce.interaction.searchAndGetScreenPopUrl(searchParams:string, queryParams:string, callType:string, callback:function)
```

引数

名前	型	説明
searchParams	string	検索対象の文字列。
queryParams	string	URL に渡すクエリパラメータを指定します。
callType	string	着信、発信、内線、または null などの通話種別を指定します。通話種別は、ソフトフォンレイアウトでの設定ごとに、語句検索を行うオブジェクトを決定します。 callType が null の場合、デフォルトで着信を検索します。callType が内線または発信の場合、画面ポップは表示されません。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/33.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      } else {
        alert(response.error);
      }
    };
    function searchAndGetScreenPopUrl () {
      //Invokes API method
      sforce.interaction.searchAndGetScreenPopUrl ('Acme',
'Key1=value1&Key2=value2', 'inbound', callback);
    }
  </script>
</head>
<body>
  <button onclick="searchAndGetScreenPopUrl ();">searchAndGetScreenPopUrl</button>
</body>
</html>
```

応答

名前	型	説明
result	string	検索結果および URL に一致するオブジェクトのリストを画面ポップ (screenPopUrl) に返します。検索は、スマートフォンレイアウトで指定されたオブジェクトで実行されます。検出されたオブジェクトごとに、オブジェクト ID、項目名、項目値、および API バージョン 33.0 以降の場合はオブジェクトタブ名が JSON 文字列として返されます。 次に、「Acme」を検索し、1つの取引先オブジェクトと3つの商談オブジェクトが検出された例を示します。 <pre>{ "006x0000001ZcyG": { "Name": "Acme - 600 Widgets", "object": "Opportunity", "displayName": "Opportunity" }, "001x0000003DGQR": { "Name": "Acme", "Type": "Analyst", "object": "Account", "displayName": "Company" }, "006x0000001ZcyH": { "Name": "Acme - 200 Widgets", "object": "Opportunity", "displayName": "Opportunity" }, "006x0000001ZcyF": { "Name": "Acme - 1,200 Widgets", "object": "Opportunity", "displayName": "Opportunity" }, screenPopUrl: "/search/SearchResults?searchType=2&str=Acme" }</pre>

名前	型	説明
		API バージョン 31.0 以降では、個人取引先オブジェクトで API メソッドを呼び出すと追加情報を返します。 <pre>{ "001D000000Jn5C5": { "Name": "PersonAccount", "contactId": "003D000000QGCeU", "Type": "Analyst", "object": "Account", "displayName": "Account", "personAccount": true }, "screenPopUrl": "/001D000000Jn5C5" }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: カスタムソフトフォンレイアウトの設計](#)

Salesforce Classic の searchAndScreenPop()

使用方法

特定の文字列に対してソフトフォンレイアウトで指定されたオブジェクトを検索します。検索結果を返し、一致するレコードが画面にポップされます。このメソッドは、ソフトフォンレイアウトで定義された画面ポップ設定を優先します。

構文

```
sforce.interaction.searchAndScreenPop(searchParams:string, queryParams:string, callType:string, (optional) callback:function);
```

引数

名前	型	説明
searchParams	string	検索対象の文字列。
queryParams	string	URL に渡すクエリパラメータを指定します。
callType	string	着信、発信、内線、または null などの通話種別を指定します。通話種別は、ソフトフォンレイアウトでの設定ごとに、語句検索を行うオブジェクトを決定します。 callType が null の場合、デフォルトで着信を検索します。callType が内線または発信の場合、画面ポップは表示されません。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/33.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      } else {
        alert(response.error);
      }
    };
    function searchAndScreenPop() {
      //Invokes API method
      sforce.interaction.searchAndScreenPop('Acme', 'Key1=value1&Key2=value2',
'inbound', callback);
    }
  </script>
</head>
<body>
  <button onclick="searchAndScreenPop();">searchAndScreenPop</button>
</body>
</html>
```

応答

名前	型	説明
result	string	検索結果に一致するオブジェクトのリストを返します。検索は、ソフトウェアレイアウトで指定されたオブジェクトで実行されます。検出されたオブジェクトごとに、オブジェクト ID、項目名、項目値、および API バージョン 33.0 以降の場合はオブジェクトタブ名が JSON 文字列として返されます。 次に、「Acme」を検索し、1つの取引先オブジェクトと3つの商談オブジェクトが検出された例を示します。

```
{
  "006x0000001ZcyG" : {"Name" : "Acme - 600 Widgets",
"object" : "Opportunity", "displayName" :
"Opportunity"},
  "001x0000003DGQR" : {"Name" : "Acme", "Type" :
"Analyst", "object" : "Account", "displayName" :
"Company"},
  "006x0000001ZcyH" : {"Name" : "Acme - 200 Widgets",
"object" : "Opportunity", "displayName" :
"Opportunity"},
  "006x0000001ZcyF" : {"Name" : "Acme - 1,200 Widgets",
"object" : "Opportunity", "displayName" : "Opportunity"}
}
```


名前	型	説明
		API バージョン 31.0 以降では、個人取引先オブジェクトで API メソッドを呼び出すと追加情報を返します。 <pre>{ "001D000000JWAW8": { "Name": "Acme", "contactId": "003D000000QNwDB", "Type": "Analyst", "object": "Account", "personAccount": true } }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: カスタムソフトフォンレイアウトの設計](#)

Salesforce Classic の setVisible()

使用方法

Salesforce コンソールで、ソフトフォンを表示するか非表示にします。

 **メモ:** このメソッドが、マルチモニターコンポーネントが有効になっている Salesforce コンソールで使用されると、エラーが返されます。

構文

```
sforce.interaction.setVisible(value:boolean, (optional) callback:function)
```

引数

名前	型	説明
value	boolean	ソフトフォンを表示するには値を true に設定し、ソフトフォンを非表示にするには値を false に設定します。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
```

```

        alert(response.result);
    } else {
        alert(response.error);
    }
};
function setVisible(value) {
    sforce.interaction.setVisible(value, callback);
}
</script>
</head>
<body>
    <button onclick="setVisible(false);">hide softphone</button>
</body>
</html>

```

応答

名前	型	説明
result	boolean	ソフトフォンの表示または非表示が正常に行われた場合は true、ソフトフォンの表示または非表示が正常に行われなかった場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: Salesforce コンソールのマルチモニターコンポーネントの有効化](#)

コンピュータテレフォニーインテグレーション (CTI) のメソッド

Open CTI を使用すると、CTI システムを Salesforce に統合できます。

重要: Open CTI を実装する方法は、組織のユーザインターフェースに応じて異なります。Salesforce Classic と Lightning Experience 用に別個の Open CTI API があります。これら 2 つの Open CTI API は動作や機能が異なるため、カスタム JavaScript コードで交換することはできません。開発を始める前に、どこに CTI システムを実装するかを必ず検討してください。

[Salesforce Classic の disableClickToDial\(\)](#)

[Salesforce Classic の enableClickToDial\(\)](#)

[Salesforce Classic の getCallCenterSettings\(\)](#)

[Salesforce Classic の getDirectoryNumbers\(\)](#)

[Salesforce Classic の getSoftphoneLayout\(\)](#)

[Salesforce Classic の onClickToDial\(\)](#)

[Salesforce Classic の setSoftphoneHeight\(\)](#)

[Salesforce Classic の setSoftphoneWidth\(\)](#)

関連トピック:

[UI が重要となる理由 — Open CTI for Salesforce Classic と Lightning Experience の比較](#)

[Salesforce Classic と Lightning Experience の Open CTI 間のメソッドの等価性](#)

Salesforce Classic の disableClickToDial()

使用方法

クリック-to-ダイヤルを無効にします。

構文

```
sforce.interaction.cti.disableClickToDial( (optional) callback: function )
```

引数

名前	タイプ	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Click to dial was disabled.');
      } else {
        alert('Click to dial was not disabled.');
      }
    };
    function disableClickToDial() {
      //Invokes API method
      sforce.interaction.cti.disableClickToDial(callback);
    }
  </script>
</head>
<body>
```

```
<button onclick="disableClickToDial();">disable click to dial</button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	boolean	クリック-to-ダイヤルが無効にされた場合は true、クリック-to-ダイヤルが無効にされなかった場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の enableClickToDial()

使用方法

クリック-to-ダイヤルを有効にします。

構文

```
sforce.interaction.cti.enableClickToDial( (optional) callback:function )
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Click to dial was enabled.');
      } else {
        alert('Click to dial was not enabled.');
      }
    }
  </script>
</head>
</html>
```

```

    }
  };
  function enableClickToDial() {
    //Invokes API method
    sforce.interaction.cti.enableClickToDial(callback);
  }
</script>
</head>
<body>
  <button onclick="enableClickToDial();">enable click to dial</button>
</body>
</html>

```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	boolean	クリック-to-ダイヤルが有効にされた場合は true、クリック-to-ダイヤルが有効にされなかった場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の getCallCenterSettings()

使用方法

コールセンター定義ファイル内のコールセンターの設定を JSON 文字列として返します。詳細は、「[コールセンター定義ファイル](#)」を参照してください。

構文

```
sforce.interaction.cti.getCallCenterSettings(callback: function)
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      alert(response.result);
    }

    //Calls getCallCenterSettings
    sforce.interaction.cti.getCallCenterSettings(callback);
  </script>
</head>
<body></body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	string	API コールが正常に行われた場合、コールセンターの設定定義が JSON 文字列として返されます。API コールが失敗した場合、null が返されます。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の getDirectoryNumbers()

使用方法

コールセンターのディレクトリから電話番号のリストを返します。このメソッドは、API バージョン 31.0 以降でのみ使用できます。

構文

```
sforce.interaction.cti.getDirectoryNumbers(isGlobal:boolean, callCenterName:String,
(optional) callback:function, (optional) resultSetPage:Integer, (optional)
resultSetPageSize:Integer)
```

引数

名前	型	説明
isGlobal	boolean	グローバルコールセンター名からディレクトリ番号を返すには値を <code>true</code> に設定し、コールセンター固有のディレクトリ番号を返すには値を <code>false</code> に設定します。
callCenterName	string	ディレクトリ番号を返すコールセンター名を指定します。isGlobal が <code>false</code> に設定されていて、この項目が指定されない場合、すべてのディレクトリ番号が返されます。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。
resultSetPage	integer	返される結果のリストのページ番号を表します。この番号は 0 から開始します。
resultSetPageSize	integer	検索する電話番号の最大数を設定します。これはデフォルトでは 5000 に設定されていて、最大数は 10000 です。hasNext が callback で <code>true</code> を返す場合、この引数を resultSetPage と一緒に使用して、結果の次のページを取得します。たとえば、resultSetPageSize が 5000 に設定されていて、resultSetPage が 0 に設定されている場合、最初の 5000 件の電話番号が返されます。resultSetPage が 1 に設定されている場合、次の 5000 件の電話番号が返されます。

サンプルコード – JavaScript

```
<html>
<head>
  <script src="https://domain:port/support/api/31.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert(response.result);
      } else {
        alert(response.error);
      }
    };

    var isGlobal = false; //Do not return directories from the global call center
    var callCenterName = 'My Call Center'; //Call center name of directory numbers to
return

    function getDirectoryNumbers() {
      sforce.interaction.cti.getDirectoryNumbers(isGlobal, callCenterName,
callback);
    }
  </script>
</head>
```

```
<body>
  <button onclick="getDirectoryNumbers();" >Get Directory Numbers</button>
</body>
</html>
```

応答

名前	型	説明
result	string	指定されたコールセンター名から電話番号のリストを表す JSON 文字列を返します。各電話番号要素には、コールセンター名、電話番号、および説明が含まれます。次に例を示します。 <pre>{ directoryNumbers: [{callCenterName:'Demo Call Center', name:'Sales Cloud', phone:'415-555-1212', description:'Sales Cloud additional number'}, {callCenterName:'Demo Call Center 2', name:'Service Cloud', phone:'415-555-3434', description:'Service Cloud additional number'},], hasNext: false }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の getSoftphoneLayout ()

使用方法

ソフトフォンレイアウトを JSON 文字列として返します。このメソッドは、API バージョン 27.0 以降でのみ使用できます。

構文

```
sforce.interaction.cti.getSoftphoneLayout (callback: function);
```

引数

名前	型	説明
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/27.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      alert(response.result);
    }
    // Calls getSoftphoneLayout
    sforce.interaction.cti.getSoftphoneLayout(callback);
  </script>
</head>
<body></body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	string	API コールが正常に行われた場合、ソフトフォンレイアウト定義が JSON 文字列として返されます。API コールが失敗した場合、null が返されます。 返された JSON 文字列には、各通話種別を表す 3 つの要素が含まれます。 "Internal" "Inbound" "Outbound" 各通話種別には 3 つのサブセクションが含まれます。 "callRelatedFields" — 表示対象として選択された通話関連項目の配列。使用できる値は、"ANI"、"DNIS"、"SEGMENT"、"QUEUE" です。 "objects" — 表示対象として選択された Salesforce オブジェクトのセット。各オブジェクトから表示対象として選択された [項目表示ラベル] および [項目名] (API 参照名) も含まれます。 "screenPopSettings" — このオブジェクトには、"ExistingWindow" または "NewWindow" のいずれかの値を持つ "screenPopsOpenWithin" 項目が含まれます。また、各画面ポップの一致項目、"NoMatch"、"SingleMatch"、"MultipleMatches" の設定も含まれます。各一致項目には、対応する "screenPopType" 項目が格納されます。また、"screenPopData" 項目が格納され

名前	型	説明
		る場合もあります。"screenPopType" に "PopToEntity" の値が指定されている場合、"screenPopData" には対象オブジェクトの名前が含まれます。"screenPopType" に "PopToVisualforce" の値が指定されている場合、"screenPopData" には対象の Visualforce ページの名前が含まれます。"screenPopType" に "PopToSearch" の値が指定されている場合、"screenPopData" 項目は存在しません。 次に、JSON 応答の例を示します。 <pre> "Internal" : { "callRelatedFields" : ["ANI", "DNIS",] "objects" : { "User" : [{ "displayName" : "Name", "apiName" : "Name" }] }, "screenPopSettings" : {} }, "Inbound" : { "callRelatedFields" : ["ANI", "DNIS", "SEGMENT", "QUEUE"], "objects" : { "Account" : [{ "displayName" : "Account Name", "apiName" : "Name" }] }, "screenPopSettings" : { "NoMatch" : { "screenPopType" : "PopToEntity", "screenPopData" : "Contact" }, "SingleMatch" : { "screenPopType" : "PopToVisualforce", "screenPopData" : "Visualforce_Page_Name" }, "MultipleMatches" : { "screenPopType" : "PopToSearch" } } </pre>

名前	型	説明
		<pre> } } }, "Outbound" : { "callRelatedFields" : ["DNIS"], "objects" : { "Account" : [{ "displayName" : "Account Name", "apiName" : "Name" }] }, "screenPopSettings" : {} } } </pre>
error	文字列型または未定義	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: カスタムソフトフォンレイアウトの設計](#)

Salesforce Classic の onClickToDial ()

使用方法

有効化された電話番号をユーザがクリックするとコールされる関数を登録します。

構文

```
sforce.interaction.cti.onClickToDial( listener: function )
```

引数

名前	型	説明
listener	function	ユーザが電話番号をクリックしたときにコールされる JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var listener = function (response) {
      if (response.result) {
        alert('User clicked on a phone number.' + response.result );
      }
    };
    //Invokes API method
    sforce.interaction.cti.onClickToDial(listener);
  </script>
</head>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	string	クリックが開始された場所から、電話番号、オブジェクト ID、オブジェクト名、および API バージョン 33.0 以降の場合はオブジェクトタブ名を JSON 文字列として返します。以下に例を示します。 <pre>{ "number": "4155551212", "objectId": "001x0000003DIGj", "objectName": "Account", "displayName": "Company" }</pre> API バージョン 33.0 以降では、個人取引先オブジェクトで API メソッドを呼び出すと次の追加情報を返します。 - AccountId または ContactId、関連する取引先または取引先責任者の ID - PersonAccount、オブジェクトが個人取引先の場合は true、それ以外の場合は false 次に例を示します。 <pre>{ "number": "4155551212", "objectId": "001D000000JWVvP", "objectName": "Howard Jones", "object": "Account", "personAccount": true, "contactId": " 003D000000QOBPX" }</pre>
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

Salesforce Classic の setSoftphoneHeight()

使用方法

ソフトフォンの高さ (ピクセル単位) を設定します。

 **メモ:** このメソッドが、マルチモニターコンポーネントが有効になっている Salesforce コンソールで使用される場合、マルチモニターコンポーネントのサイズ調整は許可されないためエラーが返されます。

構文

```
sforce.interaction.cti.setSoftphoneHeight(height:number, (optional) callback:function)
```

引数

名前	型	説明
height	number	ソフトフォンの高さ (ピクセル単位)。高さは 0 以上の数値である必要があります。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
      if (response.result) {
        alert('Height was set successfully.');
      }
      else {
        alert('Height was not set successfully.');
      }
    };
  </script>
</head>
<body>
  <button onclick="sforce.interaction.cti.setSoftphoneHeight(200, callback);">
    set softphone height to 200px
  </button>
</body>
</html>
```

応答

名前	型	説明
result	boolean	高さが正常に設定された場合は true、高さが正常に設定されなかった場合は false。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: Salesforce コンソールのマルチモニターコンポーネントの有効化](#)

Salesforce Classic の setSoftphoneWidth()

使用方法

Salesforce コンソールのソフトフォンの幅 (ピクセル単位) を設定します。

 **メモ:** このメソッドが、マルチモニターコンポーネントが有効になっている Salesforce コンソールで 사용되는場合、マルチモニターコンポーネントのサイズ調整は許可されないためエラーが返されます。

構文

```
sforce.interaction.cti.setSoftphoneWidth(width:number, (optional) callback:function)
```

引数

名前	型	説明
width	number	ソフトフォンの幅 (ピクセル単位)。幅は 0 以上の数値である必要があります。
callback	function	API メソッドコールが完了するときに実行された JavaScript メソッド。

サンプルコード – JavaScript

```
<html>
<head>
  <script type="text/javascript"
src="http://domain:port/support/api/25.0/interaction.js"></script>
  <script type="text/javascript">
    var callback = function (response) {
```

```

        if (response.result) {
            alert('Width was set successfully.');
```

```
        }
        else {
            alert('Width was not set successfully.');
```

```
        }
    };
</script>
</head>
<body>
    <button onclick="sforce.interaction.cti.setSoftphoneWidth(100, callback);">
        set softphone width to 100px
    </button>
</body>
</html>
```

応答

このメソッドは非同期です。応答は、コールバックメソッドに渡されたオブジェクトで返されます。応答オブジェクトには次の項目が含まれます。

名前	型	説明
result	boolean	幅が正常に設定された場合は <code>true</code> 、幅が正常に設定されなかった場合は <code>false</code> 。
error	string	API コールが正常に行われた場合、この変数は未定義です。API コールが失敗した場合、この変数はエラーメッセージを返します。

関連トピック:

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: Salesforce コンソールのマルチモニターコンポーネントの有効化](#)

第 6 章 その他のリソース

このガイドの他にも、以下のように、Open CTI の使用方法を学習するためのリソースが用意されています。

Open CTI 表記規則

表記規則はコード例の中で使用されます。Courier フォント、斜体、括弧などの意味を説明します。

関連トピック:

[Salesforce ヘルプ: コールセンターの概要](#)

[Salesforce ヘルプ: Salesforce コンソール](#)

[Salesforce ヘルプ: 用語集](#)

[Salesforce 開発者: Salesforce App Cloud の使用開始](#)

[Salesforce University: トレーニング](#)

Open CTI 表記規則

表記規則はコード例の中で使用されます。Courier フォント、斜体、括弧などの意味を説明します。

規則	説明
Courier font	構文の記述では、等幅フォントは、角かっこを除いて表示されたとおりに入力する必要のある項目を示します。次に例を示します。 <pre>Public class HelloWorld</pre>
斜体	構文の記述では、斜体は変数を示します。実際の値を入力してください。次の例では、3つの値を入力する必要があります。datatype variable_name [= value]; 構文で太字かつ斜体のテキストは、クラス名や変数の値など、ユーザが指定する必要があるコード要素を表します。 <pre>public static class YourClassHere { ... }</pre>
Bold Courier font	コードサンプルと構文の記述では、太字の Courier フォントはコードまたは構文の部分を強調します。

規則	説明
< >	構文の記述では、不等号 (< >) は表示されたとおりに入力します。 <pre><apex:pageBlockTable value="{!account.Contacts}" var="contact"> <apex:column value="{!contact.Name}"/> <apex:column value="{!contact.MailingCity}"/> <apex:column value="{!contact.Phone}"/> </apex:pageBlockTable></pre>
{ }	構文の説明では、中括弧 ({ }) は表示されたとおりに入力します。 <pre><apex:page> Hello {!\$User.FirstName}! </apex:page></pre>
[]	構文の記述では、角括弧で囲まれるものはすべて省略可能です。次の例では、value の指定は省略可能です。 <pre>data_type variable_name [= value];</pre>
	構文の記述では、パイプ記号は「または」を意味します。次のいずれか(すべてではない)を実行できます。次の例では、2つの方法のいずれかを使用して未入力のセットを作成するか、次のようにセットを入力することができます。 <pre>Set<data_type> set_name [= new Set<data_type>();] [= new Set<data_type{value [, value2. . .] };] ;</pre>