



## **Cisco Webex Contact Center デスクトップ開発者ガイド**

初版：2020年12月2日

最終更新：2021年1月22日

### **シスコシステムズ合同会社**

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスココンタクトセンター

0120-092-255（フリーコール、携帯・PHS含む）

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>

【注意】 シスコ製品をご使用になる前に、安全上の注意（ [www.cisco.com/jp/go/safety\\_warning/](http://www.cisco.com/jp/go/safety_warning/) ）をご確認ください。本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021 Cisco Systems, Inc. All rights reserved.



## 目次

---

第 1 章	はじめに 1
	変更履歴 1
	Agent Desktop 2
	技術概要 2

---

第 1 部 :	ウィジェット 5
---------	----------

---

第 2 章	カスタムウィジェット 7
	カスタムウィジェットの作成 7
	はじめに 8
	プロパティまたは属性インターフェイスの定義 8
	リアクティブプロパティまたは属性の変更 9
	データプロバイダー：ウィジェットのプロパティと属性 10
	Momentum UI Web コンポーネントライブラリ 11
	ライトモードまたはダークモードのサポート 13

---

第 II 部 :	JavaScript SDK とモジュール 15
----------	--------------------------

---

第 3 章	JavaScript SDK 17
	JavaScript SDK 17
	はじめに 18
	ルート JavaScript SDK モジュール 19

---

第 4 章	構成モジュール 21
	メソッド 21

[init\(accessToken, Service\)](#) 21

[clientLocale\(\)](#) 22

---

第 5 章

[ローカリゼーションモジュール](#) 23

メソッド 24

[init\(initOptions\)](#) 24

[createInstance\(createOptions\)](#) 25

[createMixin\(\)](#) 25

[getMergedInitOptions\(\)](#) 25

[cleanup\(\)](#) 26

---

第 6 章

[アクションモジュール](#) 27

メソッド 27

[getToken\(\)](#) 27

[getIdleCodes\(\)](#) 27

[getWrapUpCodes\(\)](#) 28

[getMediaTypeQueue\(telephony, social, email, chat\)](#) 28

[fireGeneralSilentNotification\(raw\)](#) 34

[fireGeneralAutoDismissNotification\(raw\)](#) 35

[fireGeneralAcknowledgeNotification\(raw\)](#) 36

[addCustomTask\(\)](#) 37

[getTaskMap\(\)](#) 40

---

第 7 章

[ロガーモジュール](#) 43

メソッド 44

[createLogger\(my-custom-component\)](#) 44

[browserDownloadLogsJson\(my-custom-component\)](#) 44

[browserDownloadLogsText\(my-custom-component\)](#) 45

[getLogsCollection\(my-custom-component\)](#) 45

[getLogsJsonUrl\(my-custom-component\)](#) 45

[getLogsTextUrl\(my-custom-component\)](#) 46

[cleanupPrefixedLogs\(my-custom-component\)](#) 46

---

第 8 章	エージェント状態情報モジュール	47
	メソッド	48
	latestData	48
	stateChange(state, auxCodeIdArray)	49
	fetchAddressBooks()	50
	イベント	50
	addEventListener	50

---

第 9 章	エージェントコンタクトモジュール	53
	メソッド	55
	accept()	55
	consultAccept()	59
	end()	63
	buddyAgents()	66
	consult()	67
	consultConference()	72
	consultEnd()	77
	decline()	79
	cancelCtq()	81
	wrapup()	83
	vTeamList()	86
	vTeamTransfer()	88
	blindTransfer()	92
	consultTransfer()	96
	hold()	100
	unHold()	103
	pauseRecording()	106
	resumeRecording()	108
	非同期イベント	111
	メソッド	113
	addEventListener(event, handler)	113
	removeEventListener(event, handler)	113

addOnceEventListener(event, handler)	113
removeOnceEventListener(event, handler)	114
removeAllEventListeners(event, handler)	114
イベント	114
eAgentContact	114
eAgentContactAssigned	116
eAgentContactAssignFailed	116
eAgentContactWrappedUp	116
eAgentOfferContact	116
eAgentOfferContactRona	119
eAgentOfferConsult	120
eAgentWrapup	122
eAgentContactHeld	122
eAgentContactUnHeld	122
eCallRecordingStarted	122
eResumeRecording	125
ePauseRecording	125
eConsultTransfer	126
eAgentblindTransferred	126
eAgentvteamTransfer	126
eAgentConsultCreated	126
eAgentConsultConferenced	126
eAgentConsultEnded	127
eAgentCtqCancelled	127
eAgentConsultConferenceEnded	127
eAgentConsulting	127
eAgentConsultFailed	127
eAgentConsultEndFailed	127
eAgentCtqFailed	128
eAgentCtqCancelFailed	128
eAgentConsultConferenceEndFailed	128

	<a href="#">startOutdial(data)</a>	130
	<a href="#">イベント</a>	132
	<a href="#">eOutdialFailed</a>	132
<hr/>		
第 11 章	<a href="#">スクリーンポップモジュール</a>	135
	<a href="#">イベント</a>	135
	<a href="#">eScreenPop</a>	135
<hr/>		
第 12 章	<a href="#">ショートカットキーモジュール</a>	137
	<a href="#">メソッド</a>	138
	<a href="#">register()</a>	138
	<a href="#">getRegisteredKeys()</a>	139
	<a href="#">デフォルトのショートカットキー</a>	143
	<a href="#">修飾子</a>	147
	<a href="#">登録済みキー</a>	147
	<a href="#">unregisterKeys()</a>	148
	<a href="#">listenKeyPress(event)</a>	148
	<a href="#">listenKeyConflict(event)</a>	149
	<a href="#">listenConflictResolved()</a>	149
<hr/>		
第 III 部 :	<a href="#">移行</a>	151
<hr/>		
第 13 章	<a href="#">Finesse ガジェットの移行</a>	153
	<a href="#">Finesse 埋め込み Web アプリケーションガジェットの移行</a>	153
	<a href="#">JavaScript ベースの Finesse ガジェットの移行</a>	155







# 第 1 章

## はじめに

- [変更履歴 \(1 ページ\)](#)
- [Agent Desktop \(2 ページ\)](#)

## 変更履歴

次の表に、このガイドで行われた変更のリストを示します。最新の変更が上部に表示されます。

変更	参照先	日付
<p>JavaScript API リファレンスは JavaScript SDK に名称変更されました。</p> <p><b>警告</b> agentx という語の参照は修正されています。</p> <ul style="list-style-type: none"><li>• @agentx/agentx および @agentx/agentx-services-types は、@wxcc-desktop/sdk-types の共通名でエクスポートされます。</li><li>• @agentx/agentx-js-api は @wxcc-desktop/sdk に名称変更されています。</li><li>• agentxJsApi は Desktop に名称変更されています</li></ul>	-	2021 年 1 月
新しいモジュール、メソッド、例が追加されました。	<a href="#">JavaScript SDK (17 ページ)</a>	

変更	参照先	日付
このマニュアルの初版		2020年12月

## Agent Desktop

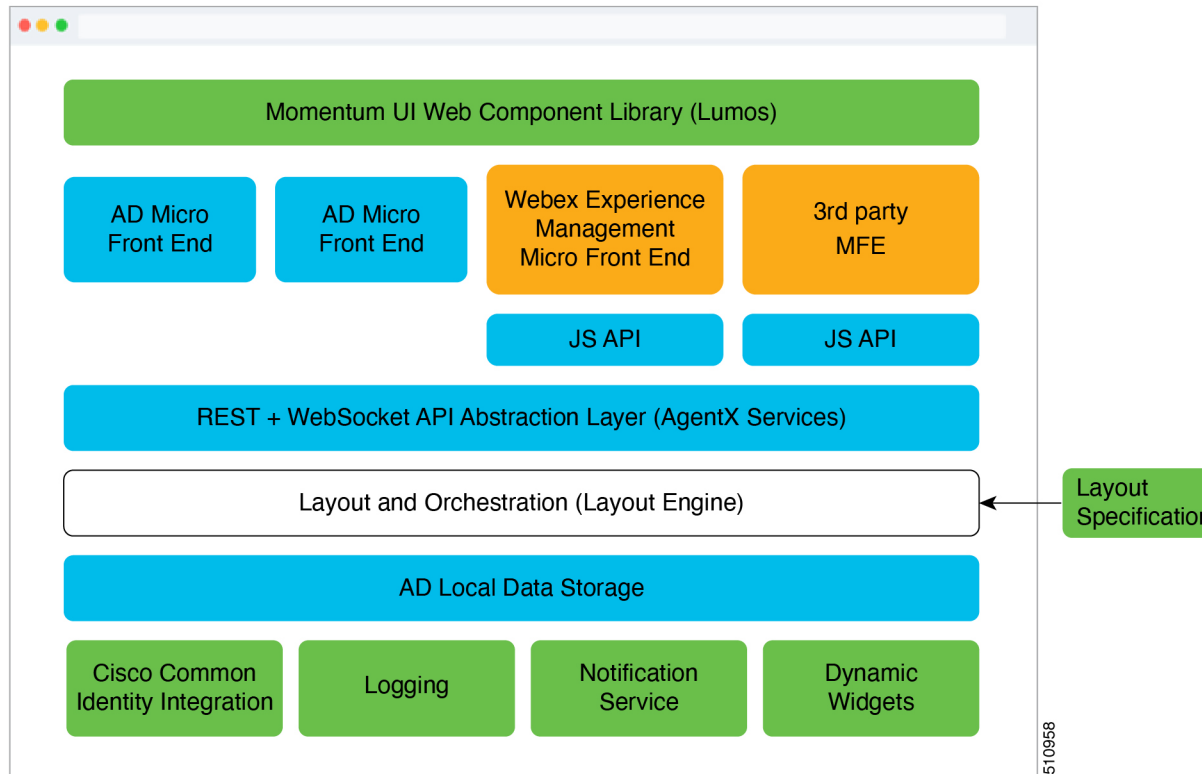
Cisco Webex Contact Center Agent Desktop は、シスコに期待されるセキュリティとグローバルなスケーラビリティにより、ビジネスにイノベーション、拡張性、クラウドのアジリティを提供します。エージェントの生産性を維持するには、効率的なプロセスと直感的なデスクトップツールが必要です。定期的なタスクの自動化が進むほど、エージェントはさらに優れたサービスを顧客に提供できます。

エージェントデスクトップは、1つの統一されたデスクトップエクスペリエンスで、音声、チャット、電子メール、ソーシャルチャット（Facebook Messenger や SMS）など、顧客とのすべてのインタラクションに対応します。CRMなどのビジネスアプリケーションを統合すれば、アプリケーションの切り替えを最小限にできます。

## 技術概要

次の図は、さまざまなマイクロフロントエンドとサービスの関係を示します。

図 1: Agent Desktopのアーキテクチャ







## 第 1 部

# ウィジェット

- [カスタムウィジェット \(7 ページ\)](#)





## 第 2 章

# カスタムウィジェット

- [カスタムウィジェットの作成 \(7 ページ\)](#)
- [はじめに \(8 ページ\)](#)
- [プロパティまたは属性インターフェイスの定義 \(8 ページ\)](#)
- [リアクティブプロパティまたは属性の変更 \(9 ページ\)](#)
- [データプロバイダー：ウィジェットのプロパティと属性 \(10 ページ\)](#)
- [Momentum UI Web コンポーネントライブラリ \(11 ページ\)](#)

## カスタムウィジェットの作成

Cisco Webex Contact Center Agent Desktopは、ウィジェットの形式で拡張機能をサポートします。ウィジェットは、デスクトップのカスタマイズに必須のコンポーネントです。ウィジェットは、特定のカプセル化された機能を備えたコンポーネントであり、デスクトップ内に配置されるカスタム HTML 要素としてエクスポートされます。より厳密には、これらのカスタム HTML 要素を Web コンポーネントにすることを推奨します。これは、どのウィジェットも最終的に Shadow Root のネストツリー内に配置されるためであり、スタイル設定のカプセル化（または何らかのスタイル設定）を有効にするためにはそれぞれ固有の Shadow Document Object Model (DOM) を有効にする必要があります。

Webex Contact Center Agent Desktopのウィジェットは Web コンポーネントまたは埋め込み iFrame である必要があるとわかると、開発者はテクノロジーを強制されるように感じるかもしれません。しかし、互換性のある任意のフレームワークでこれらのコンポーネントを構築できます。唯一の要件は、カスタムウィジェットを Web コンポーネントとしてエクスポートする必要があるという点です。これは、どのような機能も、Shadow DOM を有効にした状態で、JS バンドル内に定義される単一のカスタム HTML 要素としてラップする必要があるためです。

次に、いくつかの例を示します。

- Angular JS でウィジェットを構築する場合は、ご利用の Angular バージョンが [Angular Elements](#) (バージョン 6 で追加) をサポートしていることを確認し、アプリケーションを Angular Element としてエクスポートします。
- (エージェントデスクトップのパーツと同様に) React でアプリケーションを構築する場合は、作成する React アプリケーションをカスタム HTML 要素にラップして、Shadow DOM

を手動で有効にするか、このプロセスを自動化するオープンソースソリューションを利用します。たとえば、[Direflow](#) があります。

- Web コンポーネントの利用や Web コンポーネントとしてのエクスポートに完全に対応しているフレームワークは他にもさまざまなものがあります。参考として、広く利用されているフレームワークに対して定期的に自動テストを実行して継続的な互換性を確認している <https://custom-elements-everywhere.com> を利用できます。

ウィジェットとしてエクスポートする Web アプリケーションを構築するのに、フレームワークを使用する必要はありません。実際に、このマニュアルに記載しているすべての例で、通常の JavaScript を参照しています。

## はじめに

AgentDesktop環境の規模が拡大してもウィジェットを配置できるような設定にすることを推奨します。参考として、Cisco DevNet の GitHub コミュニティ (<https://github.com/CiscoDevNet/webex-contact-center-widget-starter>) で入手可能な定型プロジェクトのいずれかを使用使用できます。

または、制約付きのコンテナにウィジェット (Web コンポーネント) を配置して、以下の項目についてテストできるようなサンドボックス環境を用意することも推奨します。

- さまざまなコンテナサイズ (推奨サイズから全画面まで) での応答動作。
- コンテンツが X 軸または Y 軸 (スクロールバー) の方向にオーバーフローした場合のウィジェットの表示と動作に関する、オーバーフロー動作。
- ウィジェットに渡されるデータに対する反応。リアクティブ型のコンポーネントに関する実際の使用方法の詳細については、[リアクティブプロパティまたは属性の変更 \(9 ページ\)](#) を参照してください。
- Momentum UI Web コンポーネントのライブラリを使用しているかどうかに応じた、ライトモードとダークモードのサポート。

## プロパティまたは属性インターフェイスの定義

前提条件：

Web コンポーネントとカスタム HTML 要素の操作について理解しておく必要があります。詳細については、[MDN のドキュメンテーション](#) を参照してください。

カスタム要素は、HTML 要素クラスまたは HTML 要素クラスから継承するクラスを拡張できます。

例：

```
class MyCustomElement extends HTMLElement {
  constructor() {
```



```

    // Always call super first in constructor
    super();
    const shadow = this.attachShadow({
      mode: 'open'
    });

    // write element functionality in here

    ...
  }
}

```

プロパティと属性を定義するには、コンストラクタでデフォルト値を割り当てる必要があります。



(注) データ型が文字列またはブール値であるデータを受け取る場合に限り、任意のプロパティを属性として渡すことができます。詳細については、[JavaScript.Info](#) を参照してください。

例：

```

class MyCustomElement extends HTMLElement {
  constructor() {
    // Always call super first in constructor
    super();
    const shadow = this.attachShadow({
      mode: 'open'
    });

    this.width = '100px';
    this.height = '100px';
    this.active = true;

    // Data of types Array, Object, Map, Set, etc. cannot be passed as an attribute
    this.options = [];
  }
}

```

## リアクティブプロパティまたは属性の変更

エージェントデスクトップ (ウィジェット `connectedCallback()`) から属性とプロパティを使用して最新のデータを取得するには、`HTMLElement` ライフサイクルメソッドを使用します。詳細については、[MDN のドキュメンテーション](#) を参照してください。



(注) ライフサイクルメソッドが指定されていない場合、コンポーネントは、最初のレンダリング後に渡される新しいデータに反応しません。

例：

```

class MyCustomElement extends HTMLElement {
  constructor() {
    // Always call super first in constructor
    super();
  }
}

```

```

const shadow = this.attachShadow({
  mode: 'open'
});

this.width = '100px';
this.height = '100px';
this.active = true;

// Data of types Array, Object, Map, Set, etc. cannot be passed as an attribute
this.options = [];
}

attributeChangedCallback(name, oldValue, newValue) {
  console.log('Custom element attributes changed.', name);
}

...
}

```

## データプロバイダー：ウィジェットのプロパティと属性



- (注) カスタムウィジェット内のプロパティまたは属性を使用してリアルタイムデータを受信するためには、管理者がレイアウト JSON 構成で適切に割り当てる必要があります。JSON レイアウトの詳細については、『[Cisco Webex Contact Center Setup and Administration Guide \(Cisco Webex Contact Center 設定およびアドミニストレーションガイド\)](#)』の「[プロビジョニング \(Provisioning\)](#)」の章の「[デスクトップレイアウト \(Desktop Layout\)](#)」セクションを参照してください。

JavaScript SDK サブスクリバを介してデータにアクセスするほかに、プロパティまたは属性を使用してデータを渡すようにリクエストすることもできます。プロパティまたは属性の変更に応答するようにコンポーネントが作成されている場合は、信頼できる唯一の情報源であるエージェントデスクトップから、この方法で更新されたデータを常に取得します。これはデータプロバイダーと呼ばれます。

現状では、キー STORE の下にデータプロバイダーが 1 つあります。以下に、STORE キーを使用して取得可能なすべてのデータと型定義の詳細を示します。

### 例：STORE キーの詳細

```

STORE.agent.agentId: string;
STORE.agent.agentName: string;
STORE.agent.agentPhoto: string;
STORE.agent.agentProfileID: string;
STORE.agent.allowConsultToQueue: boolean;
STORE.agent.dialPlan: Service.Aqm.Configs.DialPlan;
STORE.agent.dnNumber: string;
STORE.agent.enterpriseId: string;
STORE.agent.enterpriseId: string;
STORE.agent.cadVariables: Service.Aqm.Configs.CadVariables[];
STORE.agent.channels: {
  voiceCount: number,
  chatCount: number,
}

```

```

        emailCount: number,
        socialCount: number
    };
    STORE.agent.idleCodes: Service.Aqm.Configs.Entity[];
    STORE.agent.idleStatusTimestamp: Date;
    STORE.agent.isAgentAvailableAfterOutdial: boolean;
    STORE.agent.isAdhocDialingEnabled: boolean;
    STORE.agent.isCampaignManagementEnabled: boolean;
    STORE.agent.isEndCallEnabled: boolean;
    STORE.agent.isOutboundEnabledForAgent: boolean;
    STORE.agent.isOutboundEnabledForTenant: boolean;
    STORE.agent.privacyShieldVisible: string;
    STORE.agent.profileType: string;
    STORE.agent.subStatus: string;
    STORE.agent.subStatusChangeTimestamp: Date;
    STORE.agent.teamId: string;
    STORE.agent.teamName: string;
    STORE.agent.wrapUpData: Service.Aqm.Configs.WrapupData;

```

#### 例：アプリケーションレベル

```

STORE.app.logo: string;
STORE.app.title: string;
STORE.app.darkMode: boolean;

```

#### 例：シングルサインオン (SSO)

```

STORE.auth.accessToken: string;

```

#### 例：アプリケーション通知

```

STORE.generalNotifications.isNotificationsEnabled: boolean;
STORE.generalNotifications.isSilentNotificationsEnabled: boolean;
STORE.generalNotifications.countActivated: number;
STORE.generalNotifications.countDeactivated: number;
STORE.generalNotifications.countPending: number;
STORE.generalNotifications.countAdded: number;

```

#### 例：JSON レイアウト

```

"widgets": {
  "my-component": {
    "comp": "my-custom-component",
    "properties": {
      "agentDnNumber": "${STORE.agent.dnNumber}",
      "subStatus": "${STORE.agent.subStatus}",
      "teamId": "${STORE.agent.teamId}",
    }
  }
}

```

## Momentum UI Web コンポーネントライブラリ

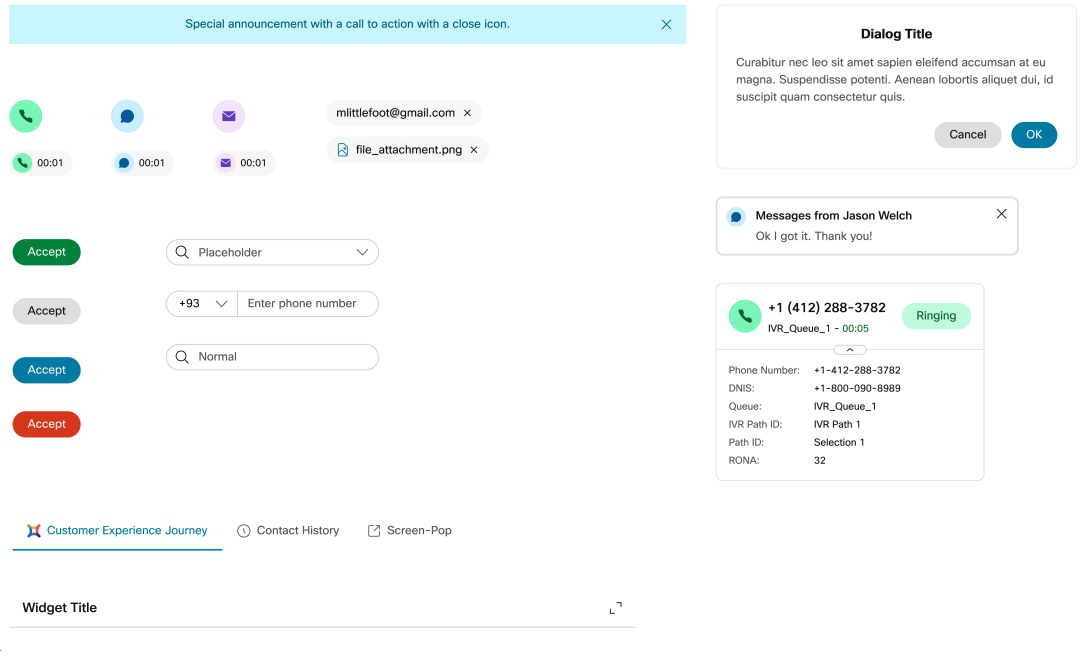
Webex Contact Center エージェントデスクトップのビジュアル言語を使用して、より短時間でカスタムウィジェットのインターフェイスを作成できるように、シスコの UI コンポーネントライブラリを確認することを推奨します。Momentum UI Web コンポーネントライブラリの詳細については、[GitHub](#) と [MIT](#) ライセンスを参照してください。問題が発生した場合は、[GitHub](#) ページで報告してください。

現在のコレクションには次のものが含まれます。

- アクティビティボタン
- アイコン
- アラート
- 入力
- アラートバナー
- ラベル
- アバター
- リンク
- バッジ
- リスト
- トピックパス(パンくずリスト)
- 項目リスト
- ボタン
- 読み込み中インジケータ
- チェックボックス
- ミーティングアラート
- 編集可能なフィールド
- メニューのオーバーレイ
- フローティングモーダル
- モーダル
- ヘルプテキスト
- 電話番号入力
- 進行状況バー (Progress Bar)
- 無線
- スピンボックス
- テーブル
- タブ
- タスク項目
- テーマ
- 切り替えスイッチ
- ヒント

Momentum UI Web コンポーネントライブラリは、今後も追加と改善が行われます。ぜひご意見をお寄せください。Web コンポーネントライブラリの詳細については、「[Components \(コンポーネント\)](#)」を参照してください。

図 2: サンプルのコンポーネント : **Agent**



## ライトモードまたはダークモードのサポート

Webex Contact Center Agent Desktopでは、デフォルトではライトモードとダークモードをサポートしています。これは、Momentum UI Web コンポーネント `<md-theme>` によって実現されます。`<md-theme>` をコンポーネントにラップすると、事前定義されたコア CSS のカラー変数と **カスタム CSS プロパティ** のコレクションに値が割り当てられます。これらのプロパティは、テーマのスイッチでライトからダークに切り替わります。CSS カスタムプロパティを利用すると、これらのプロパティは Shadow DOM を貫通し、Shadow DOM にネストされた多数のレイヤー内で使用できるという利点があります。



(注) ウィジェットの CSS が次の形式の場合、コア CSS のカラー変数を使用できます。

```
.container {
  background-color: var(--md-primary-bg-color, #fff);
}
```

### 例 : ライトモード

```
--md-default-focus-outline-color: #{$md-blue-60};

--md-primary-bg-color: #{$md-white};
--md-secondary-bg-color: #{$md-gray-05};
--md-secondary-white-bg-color: #{$md-white};
--md-tertiary-bg-color: #{$md-gray-10};
--md-tertiary-white-bg-color: #{$md-white};
--md-quaternary-bg-color: #{$md-gray-20};

--md-primary-success-bg-color: #{$md-green-10};
```

```
--md-primary-success-text-color: #{$md-green-70};

--md-primary-text-color: #{$md-gray-100};
--md-secondary-text-color: #{$md-gray-70};
--md-disabled-text-color: #{$md-gray-40};
--md-highlight-text-color: #{$md-theme-20};
--md-hyperlink-text-color: #{$md-theme-70};
--md-hyperlink-hover-text-color: #{$md-theme-90};
--md-hyperlink-focus-text-color: #{$md-theme-70};

--md-primary-seperator-color: #{$md-gray-30};
--md-secondary-seperator-color: #{$md-gray-40};

--md-presence-active-bg-color: #{$md-green-50};
--md-presence-do-not-disturb-bg-color: #{$md-red-60};
--md-presence-away-bg-color: #{$md-gray-30};
--md-presence-busy-bg-color: #{$md-yellow-40};
```

### 例：ダークモード

```
--md-default-focus-outline-color: #{$md-blue-40};

--md-primary-bg-color: #{$md-gray-100};
--md-secondary-bg-color: #{$md-gray-95};
--md-secondary-white-bg-color: #{$md-gray-95};
--md-tertiary-bg-color: #{$md-gray-90};
--md-tertiary-white-bg-color: #{$md-gray-90};
--md-quaternary-bg-color: #{$md-gray-80};

--md-primary-success-bg-color: #{$md-mint-70};
--md-primary-success-text-color: #{$md-mint-20};

--md-primary-text-color: #{$md-gray-05};
--md-secondary-text-color: #{$md-gray-40};
--md-disabled-text-color: #{$md-gray-70};
--md-highlight-text-color: #{$md-theme-80};
--md-hyperlink-text-color: #{$md-theme-40};
--md-hyperlink-hover-text-color: #{$md-theme-20};
--md-hyperlink-focus-text-color: #{$md-theme-40};

--md-primary-seperator-color: #{$md-gray-70};
--md-secondary-seperator-color: #{$md-gray-60};

--md-presence-active-bg-color: #{$md-green-50};
--md-presence-do-not-disturb-bg-color: #{$md-red-60};
--md-presence-away-bg-color: #{$md-gray-30};
--md-presence-busy-bg-color: #{$md-yellow-40};

--md-auto-wrapup-bg-color: #{$md-blue-90};
```



## 第 II 部

# JavaScript SDK とモジュール

- [JavaScript SDK \(17 ページ\)](#)
- [構成モジュール \(21 ページ\)](#)
- [ローカリゼーションモジュール \(23 ページ\)](#)
- [アクションモジュール \(27 ページ\)](#)
- [ロガーモジュール \(43 ページ\)](#)
- [エージェント状態情報モジュール \(47 ページ\)](#)
- [エージェントコンタクトモジュール \(53 ページ\)](#)
- [ダイヤラモジュール \(129 ページ\)](#)
- [スクリーンポップモジュール \(135 ページ\)](#)
- [ショートカットキーモジュール \(137 ページ\)](#)







## 第 3 章

# JavaScript SDK



### 警告

@agentxJsApi、@agentx/agentx、@agentx/agentx-services-types、@agentx/agentx-js-api などの古いライブラリは使用しないでください。

agentx という語の参照は修正されています。

- @agentx/agentx および @agentx/agentx-services-types は、@wxcc-desktop/sdk-types の共通名でエクスポートされます。
- @agentx/agentx-js-api は @wxcc-desktop/sdk に名称変更されています。
- agentxJsApi は Desktop に名称変更されています

JavaScript SDK の使用の詳細については、「はじめに」を参照してください。

- [JavaScript SDK \(17 ページ\)](#)
- [ルート JavaScript SDK モジュール \(19 ページ\)](#)

## JavaScript SDK

Agent Desktop JavaScript SDK は、npm パッケージであり、Agent Desktop から最新の情報をリクエストできます。この SDK を使用すると、エージェントの詳細、割り当てられたタスク、特定のタスクの詳細、現在のブラウザロール、シングルサインオン (SSO) 統合用の認証トークンなどの情報をリクエストできます。

SDK パッケージを使用すると、次の操作を実行できます。

- プロパティと属性を通じてウィジェットにデータを渡すようリクエストする。
- ウィジェット内部のシステムデータを使用、操作することで、より複雑な操作を実行する。
- 非同期で受信するデータをサブスクライブする。

Agent Desktop のイベントには、非同期で発生するものがあります。非同期イベントをサブスクライブし、ペイロード内のデータにアクセスするために、リスナーを追加できます。非同期イベントの例は次のとおりです。

- 新しいタスクが提示された
- 新しいタスクが割り当てられた
- 打診リクエストが作成された
- 打診が終了した
- スクリーンポップが表示された

非同期イベントの完全なリストについては、「[非同期イベント](#)」を参照してください。

## はじめに

JavaScript SDK の使用を開始するには、次のいずれかのオプションを選択できます。

- プロジェクトフォルダで次のコマンドを実行します。

```
npm install @wxcc-desktop/sdk --save
```

または

```
yarn add @wxcc-desktop/sdk
```

- 次のコマンドを実行して、`package.json` ファイルにパッケージを追加します。

```
"dependencies": {
  "@wxcc-desktop/sdk": "^1.2.2"
},
```

- **Momentum-ui Web** コンポーネントライブラリも使用する場合は、次の項目を追加する必要があります。

```
"peerDependencies": {
  "@momentum-ui/core": "19.9.2",
  "@momentum-ui/icons": "7.45.0",
  "@momentum-ui/utils": "6.2.7",
  "@momentum-ui/web-components": "^2.0.13",
  "lit-element": "2.3.1",
  "lit-html": "1.2.1"
},
```

さらに、JavaScript SDK リクエストで返される `Promise` の型定義にアクセスするためには、次のパッケージをインストールします。

```
"devDependencies": {
  "@wxcc-desktop/sdk-types": "1.0.2",
  ...
},
```



**重要** Momentum と lit-element の依存関係が `peerDependencies` に追加されます。これらの依存関係は Agent Desktop に存在しているため、2 回インポートしないでください。ウィジェットと Agent Desktop で同じバージョンを維持する方法はありません。

プロジェクトにパッケージをインストールしたら、ES6 インポートパターンに続けて適切なコンポーネントファイルにパッケージを含める必要があります。

```
import {Desktop} from "@wxcc-desktop/sdk";
```

## ルート JavaScript SDK モジュール

Desktop は、JavaScript SDK のルートモジュールです。このルートモジュールは、以下のサブモジュールへの参照を提供します。

- [構成モジュール \(21 ページ\)](#)
- [ローカリゼーションモジュール \(23 ページ\)](#)
- [アクションモジュール \(27 ページ\)](#)
- [ロガーモジュール \(43 ページ\)](#)
- [エージェント状態情報モジュール \(47 ページ\)](#)
- [エージェントコンタクトモジュール \(53 ページ\)](#)
- [ダイヤラモジュール \(129 ページ\)](#)
- [スクリーンポップモジュール \(135 ページ\)](#)
- [ショートカットキーモジュール \(137 ページ\)](#)

### 例

```
import {
  Desktop
} from "@wxcc-desktop/sdk";

const {
  config,
  i18n,
  actions,
  agentContact,
  agentStateInfo,
  dialer,
  logger,
  screenpop,
  shortcutKey,
} = Desktop;
```





## 第 4 章

# 構成モジュール

Desktop.config モジュールは、Agent Desktop の構成の詳細を提供します。

### 例

```
import {
  Desktop
} from "@wxcc-desktop/sdk";
import {
  SERVICE
} from "@wxcc-desktop/sdk-types";

await Desktop.config.init();

// After Desktop config inited, all sub-modules will inject SERVICE instance via their
init() methods automatically

// CLEANUP Desktop config is possible to re-use modules with SERVICE configured to another
environment
Desktop.config.cleanup();

// After Desktop config cleaned, all sub-modules will cleanup themselves via their
cleanup() methods automatically
```

- [メソッド \(21 ページ\)](#)

## メソッド

### init(accessToken, Service)

いずれかの Desktop モジュールを使用開始するための構成モジュールを開始します。

### 例

```
connectedCallback() {
  super.connectedCallback();
  this.init(); // Initialization the congiguration without any parameters optional
parameters are { accessToken: ACCESS_TOKEN, SERVICE: SERVICE }
}

async init() {
  await Desktop.config.init()
}
```

ウィジェットスターターの例の詳細については、[Cisco Webex Contact Center Widget Starter](#) を参照してください。

#### パラメータ

名前	タイプ	説明	必須
accessToken	文字列	クライアントには、デフォルトで暗号化されている、ユーザ ID 情報を含むアクセストークンが発行されます。これにより、保護されたリソースにアクセスできます。	オプション
サービス	文字列	ローカリゼーション、アクション、ロガーなど、Desktop モジュールが提供するサービス。	オプション

## clientLocale()

クライアントのロケールなどのシステムデータをリクエストします。

```
const locale = Desktop.config.clientLocale;
```

#### 戻り値

{String} クライアントのロケール。

#### 応答の例

```
const clientLocaleResponse = "en-US"
```



## 第 5 章

# ローカリゼーションモジュール

Desktop.i18n モジュールは、lit-element ベースのウィジェット用のローカリゼーションバンドルを作成し、維持します (Starter Widget をベースとして考慮した場合)。

ローカリゼーションモジュールは、i18next パッケージをベースとして作成されており、ウィジェット開発者が Agent Desktop の国際化メカニズムを利用し、追加のローカリゼーションバンドルを読み込むことができます。i18next の詳細については、次のリソースを参照してください。

- Desktop.i18n インスタンス化オブジェクトについては、  
<https://www.i18next.com/overview/api#instance-creation> を参照してください。
- i18n インスタンスのバックエンド構成については、  
<https://github.com/i18next/i18next-http-backend> を参照してください。
- i18n インスタンスの languageDetector 構成については、  
<https://github.com/i18next/i18next-browser-languageDetector> を参照してください。
- i18n インスタンスの初期化オプションについては、  
<https://www.i18next.com/overview/configuration-options> を参照してください。

このモジュールを使用すると、ウィジェットに lit-element および lit-html ライブラリが作成されます。

```
import {
  Desktop
} from "@wxc-cc-desktop/sdk";

...
// All CreateOptions for i18n are optional
type CreateOptions = {
  backend ? : Backend // import Backend from "i18next-http-backend";
  languageDetector ? : LanguageDetector // import LanguageDetector from
  "i18next-browser-languagedetector";
};

const i18n = Desktop.i18n.createInstance(createOptions ? : CreateOptions) // returns
instance described in https://www.i18next.com/overview/api#instance-creation
const i18nMixin = Desktop.i18n.createMixin({
  i18n /*Injecting i18n service instance into lit-element mixin */
})
```

```
// FYI you can see default options like so
console.log(Desktop.i18n.DEFAULT_INIT_OPTIONS); // => i18n.init options that are using
by Desktop by default

// To get started, Init i18n with options to be able call "t" function translations
if (!i18n.isInitialized) {
  // Here, you are adding (merging) your localization package with the Agent Desktop
  existing set of packages
  const initOptions = Desktop.i18n.getMergedInitOptions(Desktop.i18n.DEFAULT_INIT_OPTIONS
  || {}), {
    defaultNS: "my-ns", // "ns" here stands for the default JSON file name containing
    the localization
    ns: ["my-ns"],
    fallbackLng: "en",
    backend: {
      loadPath: "/.../path-to-locales/.../{lng}/{ns}.json"
    }
  });

  i18n.init(initOptions).catch(err => console.log(err));
}
```

サービスが初期化されたら、事前に作成した混合でコンポーネントを作成します。

```
import {
  customElement,
  LitElement
} from "lit-element";
import {
  html
} from "lit-html";

@customElement("my-awesome-component")
export class MyAwesomeComponent extends i18nMixin(LitElement) {
  render() {
    return html `
      <!-- i18nMixin will subscribe component tree updates on languages load & language
      change -->
      <!-- Component wrapped by i18nMixin can access t funcation via this.t(...) -->
      <p>${this.t("my-ns:key1")}</p>` <
        p > $ {
          this.t("my-ns:key2")
        } < /p>`
  }
}
```

- [メソッド \(24 ページ\)](#)

## メソッド

### init(initOptions)

ローカリゼーションモジュールを開始します。

例

```
i18n.init(initOptions)
```

パラメータ



名前	タイプ	説明	必須
initOptions	文字列	Desktopでデフォルトで使用される init オプション。 <code>console.log(Desktop.i18n.DEFAULT_INIT_OPTIONS)</code>	はい

## createInstance(createOptions)

lit-element ベースのウィジェット用にローカリゼーションバンドルを作成します。

### 例

```
const i18n = Desktop.i18n.createInstance(createOptions ? : CreateOptions)
// returns instance described in https://www.i18next.com/overview/api#instance-creation
```

### パラメータ

名前	タイプ	説明	必須
createOptions	文字列	ローカリゼーション (i18n) モジュールのオプションを作成します。	はい

## createMixin()

ローカリゼーション (i18n) サービスインスタンスを lit-element との混合として作成します。

### 例

```
const i18nMixin = Desktop.i18n.createMixin({
  i18n /*Injecting i18n service instance into lit-element mixin */
})
```

## getMergedInitOptions()

ローカリゼーションパッケージを、AgentDesktopの既存のパッケージセットに追加または統合します。

### 例

```
const initOptions = Desktop.i18n.getMergedInitOptions(Desktop.i18n.DEFAULT_INIT_OPTIONS
  || {}, {
  defaultNS: "my-ns", // "ns" here stands for the default JSON file name containing
the localization
  ns: ["my-ns"],
  fallbackLng: "en",
  backend: {
    loadPath: "/.../path-to-locales/.../{lng}/{ns}.json"
  }
});
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
DEFAULT_INIT_OPTIONS	文字列	Desktopでデフォルトで使用される init オプション。	はい
defaultNS	文字列	デフォルトのローカリゼーション JSON ファイル名。	はい
ns	文字列	ロケールの JSON ファイル名。	はい
fallbackLng	文字列	構成されたパスにファイルが見つからない場合、またはユーザーの優先言語を提供できない場合に使用する言語。	はい
backend	文字列	ロードパスが含まれます。	はい
-->loadPath	文字列	ファイルパスの場所。	はい

## cleanup()

初期化サービスが未定義の応答を返すと、クリーンアップをトリガーします。

### 例

```
cleanup() {
  this.SERVICE = undefined;

  this.logger.info("Cleaned");
}
```



## 第 6 章

# アクションモジュール

Desktop.actions モジュールは、Agent Desktop側にあるクライアント側データストアからリアルタイムデータを取得します。



(注) データプロバイダーを使用して、同じ情報をプロパティを通じてウィジェットに渡すこともできます。詳細については、「[データプロバイダー：ウィジェットのプロパティと属性](#)」を参照してください。

- [メソッド \(27 ページ\)](#)

## メソッド

### getToken()

SSO 認証に使用される認証トークンを取得します。

例

```
const accessToken = await Desktop.actions.getToken()  
// => Get current accessToken from Desktop store
```

戻り値

{String} ユーザーアクセストークン。

### getIdleCodes()

現在のアイドルコードを取得します。例：Idle、Coffee break、Meeting、Tea。

例

```
const idleCodes = await Desktop.actions.getIdleCodes()  
// => Get current idleCodes from Desktop store
```

### 戻り値

{Array} オブジェクトの配列。

### 応答の例

```
{
  id: "1643",
  isDefault: false,
  isSystem: true,
  name: "RONA"
} {
  id: "1644",
  isDefault: true,
  isSystem: false,
  name: "Meeting"
}
```

## getWrapUpCodes()

ラップアップコードを取得します。例：クレジットカードの問題、医療に関する問い合わせ、営業担当による説明。

### 例

```
const wrapUpCodes = await Desktop.actions.getWrapUpCodes()
// => Get current wrapUpCodes from Desktop store
Retrieves wrap up codes such as Ex. "Credit Card Issue", "Medical Query", "Sales Explained"
as per its configured
```

### 戻り値

{Array} オブジェクトの配列。

### 応答の例

```
{
  id: "2063",
  isDefault: false,
  isSystem: false,
  name: "Account Information Explained"
} {
  id: "2061",
  isDefault: false,
  isSystem: false,
  name: "Credit Card Issue"
}
```

## getMediaTypeQueue(telephony, social, email, chat)

現在利用可能なメディアタイプのリストを取得します。



(注) 管理者がエージェントのメディアタイプを構成します。

### 例

```
const queue = await Desktop.actions.getMediaTypeQueue("telephony" | "social" | "email"
| "chat") // => Get current media queue from Agent Desktop store
```

### パラメータ

名前	タイプ	説明	必須
telephony	文字列	音声コールのメディアチャンネルタイプ。	はい
social	文字列	メッセージングの会話でのメディアチャンネルタイプ。 サポートされるソーシャルメッセージングの会話は次のとおりです。 <ul style="list-style-type: none"> <li>• Messenger (Facebook Messenger)</li> <li>• SMS (ショートメッセージサービス)</li> </ul>	はい
email	文字列	電子メールのメディアチャンネルタイプ。	はい
chat	文字列	チャットのメディアチャンネルタイプ。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。

### 応答の例

```
{
  "key": "c9897fa7-6188-11eb-a6e4-f93bdeada4e5",
  "value": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "consultMediaResourceId": null,
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "accountId": {
          "agentEditable": false,
          "displayName": "accountId",
          "name": "accountId",
          "type": "STRING",
          "value": "e9e2c7a0-5c64-11ea-9e59-6fbf992ffc23"
        },
        "ani": {
          "agentEditable": false,
          "displayName": "ani",
          "name": "ani",
          "type": "STRING",
          "value": "janedoe@gmail.com"
        },
        "bccAddress": {
          "agentEditable": false,
          "displayName": "bccAddress",
          "name": "bccAddress",
          "type": "STRING",
          "value": ""
        },
        "ccAddress": {
          "agentEditable": false,
          "displayName": "ccAddress",

```

```

        "name": "ccAddress",
        "type": "STRING",
        "value": ""
    },
    "contentType": {
        "agentEditable": false,
        "displayName": "contentType",
        "name": "contentType",
        "type": "STRING",
        "value": "multipart/alternative"
    },
    "customerName": {
        "agentEditable": false,
        "displayName": "customerName",
        "name": "customerName",
        "type": "STRING",
        "value": "Jane Doe"
    },
    "date": {
        "agentEditable": false,
        "displayName": "date",
        "name": "date",
        "type": "STRING",
        "value": "Wed, 20 Jan 2021 06:09:06 +0000"
    },
    "dn": {
        "agentEditable": false,
        "displayName": "dn",
        "name": "dn",
        "type": "STRING",
        "value": "agent2021@gmail.com"
    },
    "entryPointId": {
        "agentEditable": false,
        "displayName": "entryPointId",
        "name": "entryPointId",
        "type": "STRING",
        "value": "AXCZuH9MXrf9I0XFBIfL"
    },
    "from": {
        "agentEditable": false,
        "displayName": "from",
        "name": "from",
        "type": "STRING",
        "value": "Jane Doe"
    },
    "fromAddress": {
        "agentEditable": false,
        "displayName": "fromAddress",
        "name": "fromAddress",
        "type": "STRING",
        "value": "janedoe@gmail.com"
    },
    "inReplyTo": {
        "agentEditable": false,
        "displayName": "inReplyTo",
        "name": "inReplyTo",
        "type": "STRING",
        "value": ""
    },
    "messageId": {
        "agentEditable": false,
        "displayName": "messageId",
        "name": "messageId",

```

```
    "type": "STRING",
    "value": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com"
  },
  "queueType": {
    "agentEditable": false,
    "displayName": "queueType",
    "name": "queueType",
    "type": "STRING",
    "value": "tam"
  },
  "reasonCode": {
    "agentEditable": false,
    "displayName": "reasonCode",
    "name": "reasonCode",
    "type": "STRING",
    "value": "Email_Queue"
  },
  "references": {
    "agentEditable": false,
    "displayName": "references",
    "name": "references",
    "type": "STRING",
    "value": ""
  },
  "replyToAddress": {
    "agentEditable": false,
    "displayName": "replyToAddress",
    "name": "replyToAddress",
    "type": "STRING",
    "value": "janedoe@gmail.com"
  },
  "ronaTimeout": {
    "agentEditable": false,
    "displayName": "ronaTimeout",
    "name": "ronaTimeout",
    "type": "STRING",
    "value": "30"
  },
  "subject": {
    "agentEditable": false,
    "displayName": "subject",
    "name": "subject",
    "type": "STRING",
    "value": "sales"
  },
  "threadId": {
    "agentEditable": false,
    "displayName": "threadId",
    "name": "threadId",
    "type": "STRING",
    "value": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com"
  },
  "toAddress": {
    "agentEditable": false,
    "displayName": "toAddress",
    "name": "toAddress",
    "type": "STRING",
    "value": "agent2021@gmail.com"
  },
  "virtualTeamName": {
    "agentEditable": false,
    "displayName": "virtualTeamName",
    "name": "virtualTeamName",
    "type": "STRING",
```

```

        "value": "Email_Queue"
    }
},
"callAssociatedDetails": {
    "accountId": "e9e2c7a0-5c64-11ea-9e59-6fbf992ffc23",
    "ani": "janedoe@gmail.com",
    "bccAddress": "",
    "ccAddress": "",
    "contentType": "multipart/alternative",
    "customerName": "Jane Doe",
    "date": "Wed, 20 Jan 2021 06:09:06 +0000",
    "dn": "agent2021@gmail.com",
    "entryPointId": "AXCZuH9MXrf9IOXFBIfl",
    "from": "Jane Doe",
    "fromAddress": "janedoe@gmail.com",
    "inReplyTo": "",
    "messageId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",
    "queueType": "tam",
    "reasonCode": "Email_Queue",
    "references": "",
    "replyToAddress": "janedoe@gmail.com",
    "ronaTimeout": "30",
    "subject": "sales",
    "threadId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",
    "toAddress": "agent2021@gmail.com",
    "virtualTeamName": "Email_Queue"
},
"callFlowParams": {
    "Automation_Email_Queue": {
        "description": "",
        "name": "Automation_Email_Queue",
        "qualifier": "vteam",
        "value": "3281",
        "valueDataType": "string"
    },
    "Dont_Use_Email": {
        "description": "",
        "name": "Dont_Use_Email",
        "qualifier": "vteam",
        "value": "3276",
        "valueDataType": "string"
    },
    "Email_Queue": {
        "description": "",
        "name": "Email_Queue",
        "qualifier": "vteam",
        "value": "3266",
        "valueDataType": "string"
    },
    "Mail_Automation_Queue": {
        "description": "",
        "name": "Mail_Automation_Queue",
        "qualifier": "vteam",
        "value": "3432",
        "valueDataType": "string"
    }
},
"callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3266",
    "accountId": "e9e2c7a0-5c64-11ea-9e59-6fbf992ffc23",
    "ani": "janedoe@gmail.com",
    "bccAddress": "",
    "ccAddress": "",

```



```

    "contentType": "multipart/alternative",
    "customerName": "Jane Doe",
    "date": "Wed, 20 Jan 2021 06:09:06 +0000",
    "dnis": "agent2021@gmail.com",
    "entryPointId": "AXCZuH9MXrf9I0XFBIfL",
    "from": "Jane Doe",
    "fromAddress": "janedoe@gmail.com",
    "inReplyTo": "",
    "messageId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "queueType": "tam",
    "reasonCode": "Email_Queue",
    "references": "",
    "replyToAddress": "janedoe@gmail.com",
    "ronaTimeout": "30",
    "subject": "sales",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "threadId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",
    "toAddress": "agent2021@gmail.com",
    "virtualTeamName": "Email_Queue",
    "vteamId": "AXCZuH9MXrf9I0XFBIfL"
  },
  "contactDirection": {
    "type": "INBOUND"
  },
  "currentVTeam": "3266",
  "interactionId": "c9897fa7-6188-11eb-a6e4-f93bdeada4e5",
  "isFcManaged": false,
  "isTerminated": false,
  "media": {
    "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com": {
      "holdTimestamp": null,
      "isHold": false,
      "mType": "mainCall",
      "mediaMgr": "emm",
      "mediaResourceId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",

      "mediaType": "email",
      "participants": [
        "janedoe@gmail.com",
        "7c867aa9-ec768-341a-b767-e5hd6ae7g701"
      ]
    }
  },
  "mediaChannel": "email",
  "mediaType": "email",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": null,
  "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
  "participants": {
    "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
      "channelId": "a72a75f8-fd30-44f9-85f1-39d633055475",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "8895579172",
      "hasJoined": true,
      "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1612248681335,
      "lastUpdated": 1612248681336,
      "name": "John Doe",
    }
  }
}

```

```

        "pType": "Agent",
        "queueId": "3266",
        "queueMgrId": "aqm",
        "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
        "siteId": "472",
        "teamId": "960",
        "teamName": "Email_Team",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "janedoe@gmail.com": {
        "id": "janedoe@gmail.com",
        "pType": "Customer",
        "type": "Customer"
    }
},
"previousVTeams": [],
"state": "connected",
"workflowManager": null
},
"interactionId": "c9897fa7-6188-11eb-a6e4-f93bdeada4e5",
"isConsulted": false,
"mediaResourceId": "289DBEAD-0715-4878-A0B9-365555C18E72@cisco.com",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "2e30a830-6611-11eb-bbe4-81e078594d7a",
"type": "AgentContact"
}
}
}

```

## fireGeneralSilentNotification(raw)

サイレント通知をトリガーします。サイレント通知はデスクトップには表示されませんが、通知センターに表示されます。[通知センター (Notification Center) ] アイコンは、未読メッセージ数の増加を示します。

### 例

```

import {
  Desktop
} from "@wxcc-desktop/sdk";
import {
  Notifications
} from "@uuiplatform/unified-ui-platform-sdk";

...

const raw: Notifications.ItemMeta.Raw = {
  data: {
    type: Notifications.ItemMeta.Type.Info,
    mode: Notifications.ItemMeta.Mode.Silent, // Change type here based on the method.

    title: "Info - Silent",
    data: "Lorem Ipsum Dolor",
  },
};

// Agent Desktop General Notifications:
Desktop.actions.fireGeneralSilentNotification(raw)
// => Fires silent notification in Agent Desktop.
// Silent notification will not have any appearance on desktop but a notification icon
will have one count increased.

```

## パラメータ

名前	タイプ	説明	必須
raw	オブジェクト	ユーザーに送信される通知のタイプに関する情報が含まれます。	はい
-->data	オブジェクト	特定の通知のオプション。	はい
--> type	列挙体	Agent Desktopに表示される通知のタイプ。 <ul style="list-style-type: none"> <li>• 情報 (Info)</li> <li>• 警告 (Warn)</li> <li>• エラー (Error)</li> <li>• 成功 (Success)</li> <li>• チャット (Chat)</li> <li>• デフォルト (Default)</li> </ul>	はい
-->mode	列挙体	メソッドに基づく通知で使用可能なモード。 <ul style="list-style-type: none"> <li>• サイレント (Silent)</li> <li>• 自動的に無視 (AutoDismiss)</li> <li>• 確認応答あり (Acknowledge)</li> </ul>	はい
->title	文字列	通知のタイトル。	はい
-->data	オブジェクト	通知の内容。	はい

**fireGeneralAutoDismissNotification(raw)**

自動却下通知をトリガーします。通知の解決済みステータス、理由、モードを返します。



- (注) Agent Desktopの通知をユーザーが無効にしている場合は、サイレント通知に変換されて mode に反映されます。

## 例

```
import {
  Desktop
} from "@wxcc-desktop/sdk";
import {
  Notifications
```

## fireGeneralAcknowledgeNotification(raw)

```

} from "@uui/unified-ui-platform-sdk";

...

const raw: Notifications.ItemMeta.Raw = {
  data: {
    type: Notifications.ItemMeta.Type.Info,
    mode: Notifications.ItemMeta.Mode.AutoDismiss, // Change type here based on the
    method.
    title: "Info - AutoDismiss",
    data: "Lorem Ipsum Dolor",
  },
};

// Unlike silent notification, auto-dismiss and acknowledge can have controlled responses,
// that may reflect in the notification status, e.g.:

// => Fires auto-dismiss notification in Agent Desktop. Returns notification resolved
// status, reason, mode.
NOTE: if Agent Desktop notifications are disabled by the user - it will be converted
into silent notification and reflected in "mode"
const [ status, reason, mode ]: [
  Notifications.ItemMeta.Status, Notifications.ItemMeta.StatusChangeEventReason,
  Notifications.ItemMeta.Mode ] = await
  Desktop.actions.fireGeneralAutoDismissNotification(raw)

```

## パラメータ

名前	タイプ	説明	必須
raw	オブジェクト	ユーザーに送信される通知のタイプに関する情報が含まれます。  詳細については、 <a href="#">fireGeneralSilentNotification(raw)</a> メソッドのパラメータ表を参照してください。	はい

## fireGeneralAcknowledgeNotification(raw)

確認通知をトリガーします。通知の解決済みステータス、理由、モードを返します。



(注) Agent Desktopの通知をユーザーが無効にしている場合は、サイレント通知に変換されて mode に反映されます。

## 例

```

import {
  Desktop
} from "@wxcc-desktop/sdk";
import {
  Notifications
} from "@uui/unified-ui-platform-sdk";

...

const raw: Notifications.ItemMeta.Raw = {
  data: {

```

```

        type: Notifications.ItemMeta.Type.Info,
        mode: Notifications.ItemMeta.Mode.Acknowledge, // Change type here based on the
method.
        title: "Info - Acknowledge",
        data: "Lorem Ipsum Dolor",
    },
};

```

```

// => Fires acknowledge notification in Agent Desktop. Returns notification resolved
status, reason, mode.
NOTE: if Agent Desktop notifications are disabled by the user - it will be converted
into silent notification and reflected in "mode"
const [ status, reason, mode ]: [ Notifications.ItemMeta.Status,
Notifications.ItemMeta.StatusChangeEventReason, Notifications.ItemMeta.Mode ] = await
Desktop.actions.fireGeneralAcknowledgeNotification(raw)

```

### パラメータ

名前	タイプ	説明	必須
raw	オブジェクト	ユーザーに送信される通知のタイプに関する情報が含まれます。  詳細については、 <a href="#">fireGeneralSilentNotification(raw)</a> メソッドのパラメータ表を参照してください。	はい

## addCustomTask()

ウィジェット内に新しいカスタムタスクを追加します。このタスクはAgent Desktopのタスクリストに表示されます。



- (注) ウィジェット内のタスクのリストを最新の状態に保つためには、イベントリスナーを追加して非同期イベントをサブスクライブする必要があります。これらのイベントは、新しいタスクがエージェントに割り当てられることを意味します。詳細については、「[非同期イベント](#)」を参照してください。

### 例

```

Desktop.actions.addCustomTask({
  export const contactPayload = {
    mediaResourceId: "49bcf26b-ec75-4351-89fa-55d54682c20c",
    eventType: "RoutingMessage",
    agentId: "16839506-7c48-4a71-ba1b-d585e5d37607",
    trackingId: "354deae-c98a-4a8c-9e04-e56e129fdb9f",
    interaction: {
      isFcManaged: false,
      isTerminated: false,
      mediaType: "telephony",
      previousVTeams: ["AXCZ0YCAXrf9I0XFBI7b"],
      state: "connected",
      currentVTeam: "3268",
      participants: {
        "+19997770096": {
          id: "+19997770096",

```

```

        pType: "Customer",
        type: "Customer"
    },
    "aad323de-32d8-48c9-af6b-b68dfbabfe20": {
        name: "uui-agent2",
        pType: "Agent",
        teamName: "Team X",
        lastUpdated: 1597681728863,
        teamId: "962",
        joinTimestamp: 1597681728863,
        isConsulted: false,
        hasJoined: true,
        consultTimestamp: null,
        dn: "9997770194",
        queueId: "3268",
        id: "16839506-7c48-4a71-balb-d585e5d37607",
        sessionId: "319d8e83-2a75-4ef3-9cb8-1c845edb15a0",
        consultState: null,
        queueMgrId: "aqm",
        siteId: "473",
        type: "Agent",
        channelId: "ab8b3603-84a7-4f61-b987-9044b4d69cf2",
        wrapUpTimestamp: null,
        isWrapUp: false
    }
},
interactionId: "49bcf26b-ec75-4351-89fa-55d54682c20c",
orgId: "f111e3af-1a45-42ef-9erf-4562354b8a25",
callAssociatedData: {
    customerName: {
        value: "test",
        name: "x",
        agentEditable: false,
        type: "s",
        displayName: "d"
    }
},
callAssociatedDetails: {
    virtualTeamName: "Queue - Telephony",
    ani: "+19997770096",
    ronaTimeout: "30",
    dn: "+12147659000",
    pathId: " StartCall PlayDone out out out",
    IvrPath: " EOI",
    subject: "",
    toAddress: "",
    inReplyTo: "",
    customerName: "",
    ccAddress: "",
    entryPointId: "",
    accountId: "",
    reasonCode: "",
    reason: "",
    references: "",
    contentType: "",
    date: "",
    replyToAddress: "",
    fromAddress: "",
    messageId: "",
    from: "",
    threadId: "",
    bccAddress: "",
    queueType: "",
    dnis: "",

```

```
        category: "",
        sourceNumber: "",
        sourcePage: "",
        appUser: "",
        customerNumber: ""
    },
    callProcessingDetails: {
        QMgrName: "aqm",
        pauseResumeEnabled: "true",
        taskToBeSelfServiced: "false",
        ani: "+19997770096",
        recordInProgress: "true",
        pauseDuration: "10",
        dnis: "+12147659000",
        tenantId: "133",
        QueueId: "3268",
        vteamId: "3268",
        jsriptId: "AXCZ4c3mjkwgAuS7vSIU",
        customerName: "",
        virtualTeamName: "Queue - Telephony",
        ronaTimeout: "30",
        category: "",
        reason: "",
        sourceNumber: "",
        sourcePage: "",
        appUser: "",
        customerNumber: "",
        reasonCode: "",
        IvPath: "EOI",
        pathId: " StartCall PlayDone out out out",
        fromAddress: ""
    },
    media: {
        "49bcf26b-ec75-4351-89fa-55d54682c20c": {
            mediaResourceId: "49bcf26b-ec75-4351-89fa-55d54682c20c",
            mediaType: "telephony",
            mediaMgr: "vmm",
            participants: ["+19997770096", "16839506-7c48-4a71-ba1b-d585e5d37607"],

            mType: "mainCall",
            isHold: false,
            holdTimestamp: null
        }
    },
    owner: "16839506-7c48-4a71-ba1b-d585e5d37607",
    mediaChannel: "broadcloud",
    contactDirection: {
        type: "INBOUND"
    },
    callFlowParams: {
        Play2: {
            name: "Play2",
            qualifier: "",
            description: "(A valid text.)",
            valueDataType: "string",
            value: "Welcome to Agent Team Space"
        },
        Queue3: {
            name: "Queue3",
            qualifier: "vteam",
            description: "(vteam, A valid VTeam.)",
            valueDataType: "string",
            value: "3268"
        }
    }
}
```

```

    }
  },
  interactionId: "49bcf26b-ec75-4351-89fa-55d54682c20c",
  orgId: "f111e3af-1a45-42ef-9erf-4562354b8a25",
  queueMgr: "aqm",
  type: "AgentContactAssigned",
  destAgentId: "16839506-7c48-4a71-ba1b-d585e5d37607",
  consultMediaResourceId: "",
  owner: "",
  isConferencing: false
});
})
// => Add custom task object in Desktop store

```

## getTaskMap()

特定の時点でエージェントに割り当てられているタスクの全リストを取得します。

### 例

```

const currentTaskMap = await Desktop.actions.getTaskMap()
// => Get current task map from Desktop store

```

### 戻り値

{Map} 結果のリスト。

### 応答の例

```

Desktop Store TaskMap:
  Map(1) {
    "851a8e81-6150-11eb-9a47-f1ca3756d4bd" => {
      ..., "eed4b53a-614f-11eb-9a47-8be6439707ee" => {
        ...
      }
    }
  [
    [Entries]
  ] {
    "851a8e81-6150-11eb-9a47-f1ca3756d4bd" => Object
  }
  key: "851a8e81-6150-11eb-9a47-f1ca3756d4bd"
  value:
    agentId: "7c867aa9-ec768-341a-b767-e5hd6ae7g701"
    eventType: "RoutingMessage"
    interaction: {
      callAssociatedData: {
        ...
      },
      callAssociatedDetails: {
        ...
      },
      callFlowParams: {
        ...
      },
      callProcessingDetails: {
        ...
      },
      contactDirection: {
        ...
      },
      ...
    }
    interactionId: "851a8e81-6150-11eb-9a47-f1ca3756d4bd"
    mediaResourceId: "4D44FF32-A585-4DD8-8422-4920FC97066A@cisco.com"
    orgId: "f111e3af-1a45-42ef-9erf-4562354b8a25"
    queueMgr: "aqm"

```



```
trackingId: "69773a10-6163-11eb-9fb9-8dadf7aee433"  
type: "AgentContactAssigned"
```





## 第 7 章

# ロガーモジュール

Desktop.logger モジュールは、サードパーティ製ウィジェット用にクライアント側のログメッセージを作成し、保持します。

個別に設定されたロガーを使用することで、（テレメトリと問題レポートに使用される）Agent Desktopのロギングシステムと統合すると同時に、明確な発生源の定義を保持することができます。ウィジェット用のロガーインスタンスは、プロジェクトごとに1回だけ作成することを推奨します。utils.ts ファイルにロガーインスタンスを作成して、コンポーネント全体で再利用することができます。

### 例

```
// utils.ts file
import {
  Desktop
} from "@wxcc-desktop/sdk";

export const logger = Desktop.logger.createLogger("my-custom-component");

// Component.ts file
import {
  logger
} from "./utils.ts";

logger.info("Info test"); // logger.info => 2020-12-16 13:11:04:971["my-custom-component",
  "Info test"]
logger.warn("Warn test"); // logger.info => 2020-12-16 13:11:04:971["my-custom-component",
  "Warn test"]
logger.error("Error test"); // logger.info => 2020-12-16
13:11:04:971["my-custom-component", "Error test"]
```

ウィジェットスターターの例の詳細については、[Cisco Webex Contact Center Widget Starter](#) を参照してください。

ログは、JSONフォーマットか、ご利用のウィジェット専用のダウンロード可能な\*.logファイルとして取得できます。ログをダウンロードするのに利用できるオプションは次のとおりです。

```
// Download logs as a JSON file for "my-custom-component" prefix:
Desktop.logger.browserDownloadLogsJson("my-custom-component");

// Download logs as a Text file for "my-custom-component" prefix:
Desktop.logger.browserDownloadLogsText("my-custom-component");
```

```
// Get logs as Objects collection for "my-custom-component" prefix:
Desktop.logger.getLogsCollection("my-custom-component");

// Get logs as base64 encoded url ready to put into link href to initiate browser download
// as a JSON file for "my-custom-component" prefix:
Desktop.logger.getLogsJsonUrl("my-custom-component");

// Get logs as base64 encoded url ready to put into link href to initiate browser download
// as a Text file for "my-custom-component" prefix:
Desktop.logger.getLogsTextUrl("my-custom-component");

// Cleanup logs from Local Storage for "my-custom-component" prefix:
Desktop.logger.cleanupPrefixedLogs("my-custom-component");
```

Agent Desktopからログをダウンロードすることもできます。詳細については、『[Cisco Webex Contact Center Agent Desktop User Guide \(Cisco Webex Contact Center エージェントデスクトップ ユーザーガイド\)](#)』の「Working with Agent Desktop (エージェントデスクトップの操作)」の章の「Download Error Report (エラーレポートのダウンロード)」セクションを参照してください。

- [メソッド \(44 ページ\)](#)

## メソッド

### createLogger(my-custom-component)

関連情報を特定のコンポーネントにログを記録するのに役立つ、サードパーティ製ウィジェット用のロガーを初期化します。ログのタイプによって異なる方法が使用されます。情報、警告、エラーなどのメッセージは、ブラウザのコンソールに記録されます。

#### 例

```
const loggerOne = Desktop.logger.createLogger("my-custom-component-one");
const loggerTwo = Desktop.logger.createLogger("my-custom-component-two");

loggerOne.info("Info test"); // console.log => "my-custom-component: Info:test"
loggerTwo.warn("Warn test"); // console.log => "my-custom-component: Warn:test"
loggerOne.error("Error test"); // console.log => "my-custom-component: Error:test"
```

#### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

### browserDownloadLogsJson(my-custom-component)

my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

#### 例

```
Desktop.logger.browserDownloadLogsJson("my-custom-component");
```

#### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

## browserDownloadLogsText(my-custom-component)

ログをテキストファイルとしてダウンロードします。my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

#### 例

```
Desktop.logger.browserDownloadLogsText("my-custom-component");
```

#### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

## getLogsCollection(my-custom-component)

オブジェクトのコレクションとしてログを取得します。my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

#### 例

```
Desktop.logger.getLogsCollection("my-custom-component");
```

#### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

## getLogsJsonUrl(my-custom-component)

Base64 エンコードされた URL としてログを取得し、ブラウザでのダウンロードを JSON ファイルとして開始します。my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

#### 例

```
Desktop.logger.getLogsJsonUrl("my-custom-component");
```

#### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

## getLogsTextUrl(my-custom-component)

Base64 エンコードされた URL としてログを取得し、ブラウザでのダウンロードをテキストファイルとして開始します。my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

### 例

```
Desktop.logger.getLogsTextUrl("my-custom-component");
```

### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい

## cleanupPrefixedLogs(my-custom-component)

ローカルストレージからログをクリーンアップします。my-custom-component プレフィックスについて、ログを JSON ファイルとしてダウンロードします。

### 例

```
Desktop.logger.cleanupPrefixedLogs("my-custom-component");
```

### パラメータ

名前	タイプ	説明	必須
my-custom-component	文字列	コンポーネントの名前。例：電子メールコンポーネント、チャットコンポーネント。	はい



## 第 8 章

# エージェント状態情報モジュール

Desktop.agentStateInfo モジュールは、エージェント関連の情報における最新のデータ更新をリッスンします。

### 例

```
import {
  Service
} from "@wxcc-desktop/sdk-types";

type LatestInfoData = {
  teamId ? : string;
  teamName ? : string;
  dn ? : string;
  status ? : string;
  subStatus ? : string;
  idleCodes ? : Service.Aqm.Configs.Entity[];
  wrapupCodes ? : Service.Aqm.Configs.Entity[];
  outDialRegex ? : string;
  isOutboundEnabledForTenant ? : boolean;
  isOutboundEnabledForAgent ? : boolean;
};
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
teamId	文字列	チームの一意の識別子。	はい
teamName	文字列	チームの名前。	はい
dn	文字列	エントリポイントに対するダイヤル番号 (DN)。	はい
status	文字列	エージェントのログインステータス (例: LoggedIn)。	はい
subStatus	文字列	エージェントの対応可能ステータス (例: Available)。	はい

名前	タイプ	説明	必須
idleCodes	文字列	ルーティングされたリクエストをエージェントが受け付ける準備が整っていないことを示すアイドル理由コード。例：Idle、Coffee break、Meeting、Tea	はい
wrapupCodes	文字列	エージェントが顧客とのインタラクションを終了したことを示すラップアップ理由コード。	はい
outDialRegex	文字列	アウトダイヤルの正規表現の有効性。	はい
isOutboundEnabledForTenant	ブール値	テナントに対してアウトバウンド機能を有効にするかどうかを決定します。  <ul style="list-style-type: none"> <li>• True : テナントに対してアウトバウンド機能を有効にします。</li> <li>• False : テナントに対してアウトバウンド機能を無効にします。</li> </ul>	はい
isOutboundEnabledForAgent	ブール値	エージェントに対してアウトバウンド機能を有効にするかどうかを決定します。  <ul style="list-style-type: none"> <li>• True : エージェントに対してアウトバウンド機能を有効にします。</li> <li>• False : エージェントに対してアウトバウンド機能を無効にします。</li> </ul>	はい

- [メソッド \(48 ページ\)](#)
- [イベント \(50 ページ\)](#)

## メソッド

### latestData

最新のユーザー関連情報を取得します。

例

```
const latestData: LatestInfoData = Desktop.agentStateInfo.latestData;
```

戻り値

{Object} 最新のユーザー関連情報をリストします。



### 応答の例

```
{
  "agentName": "John Doe",
  "teamName": "Sales",
  "teamId": "964",
  "dn": "8895579172",
  "status": "LoggedIn",
  "subStatus": "Idle",
  "idleCodes": [{
    "id": "1646",
    "isDefault": false,
    "isSystem": true,
    "name": "Aux on Login"
  }],
  "wrapupCodes": [{
    "id": "2065",
    "isDefault": false,
    "isSystem": false,
    "name": "Wrapping up as customer disconnected"
  }],
  "outDialRegex": "([0-9a-zA-Z]+[-._])*[0-9a-zA-Z]+",
  "isOutboundEnabledForTenant": true,
  "isOutboundEnabledForAgent": true
}
```

## stateChange(state, auxCodeIdArray)

ユーザーの状態の変更を取得します。

### 例

```
const state = await Desktop.agentStateInfo.stateChange({ state: s, auxCodeIdArray: "0"
});
```

### パラメータ

名前	タイプ	説明	必須
state	文字列	現在のエージェントの状態。	はい
auxCodeIdArray	文字列	エージェントの状態を表す一意の識別子。	はい

### 戻り値

{Object} エージェントの状態の変更に関する情報をリストします。

### 応答の例

```
{
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "agentSessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
    "auxCodeId": "0",
    "connectedChannels": [
      "38136791-cf79-4663-a5d0-39b0d094abf8"
    ],
    "eventType": "AgentDesktopMessage",
    "lastIdleCodeChangeTimestamp": null,
    "lastStateChangeReason": ""
  }
}
```

```

        "lastStateChangeTimestamp": 1612245463183,
        "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
        "status": "LoggedIn",
        "subStatus": "Available",
        "trackingId": "9008a4a0-651b-11eb-a97d-0d76c582d799",
        "type": "AgentStateChangeSuccess"
    },
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "trackingId": "notifs_aa848b62-4fcf-4d4c-95ad-1233bb7f899c",
    "type": "AgentStateChange"
}

```

## fetchAddressBooks()

アドレス帳の詳細を取得します。

### 例

```
const books = await Desktop.agentStateInfo.fetchAddressBooks()
```

### 戻り値

{Array} オブジェクトの配列。

### 応答の例

```

type AddressBooks = {
  speedDials: [{
    desc: 'Jane Doe',
    dn: '9997770094'
  },
  {
    desc: 'John Doe',
    dn: '9997770095'
  },
]
};
type Address = {
  desc: string;
  dn: string;
  phoneBookName ? : string;
};

```

## イベント

### addEventListener

dn、status、subStatus のいずれかのフィールドが変更された場合にログに記録される、updated という名前のイベントをリッスンします。

詳細については、[addEventListener\(event, handler\)](#) を参照してください。

### 例

```
Desktop.agentStateInfo.addEventListener("updated", updatedList =>
  console.log(updatedList) [{
    "name": "dn",

```

```
        "value": "+12580258011",  
        "oldValue": ""  
    }, {  
        "name": "status",  
        "value": "LoggedIn",  
        "oldValue": "DefaultState"  
    }, {  
        "name": "subStatus",  
        "value": "Available",  
        "oldValue": ""  
    }  
    ] *  
    /  
    )
```





## 第 9 章

# エージェントコンタクトモジュール

Desktop.agentContact モジュールは、新規タスクの到着など、agent-contact エンティティに関連する通知イベントについて、リクエストを作成し、リッスンします。

### 例

```
import {
    Desktop
} from "@wxcc-desktop/sdk";

//...

/*
    Supposing Desktop.config.init({...}) was called
*/

// List of available agent-contact aqm reqs:
await Desktop.agentContact.accept({
    ...
});
await Desktop.agentContact.consultAccept({
    ...
});
await Desktop.agentContact.buddyAgents({
    ...
});
await Desktop.agentContact.end({
    ...
});
await Desktop.agentContact.consultEnd({
    ...
});
await Desktop.agentContact.cancelCtq({
    ...
});
await Desktop.agentContact.wrapup({
    ...
});
await Desktop.agentContact.vteamTransfer({
    ...
});
await Desktop.agentContact.blindTransfer({
    ...
});
await Desktop.agentContact.hold({
    ...
});
```

```

await Desktop.agentContact.unHold({
  ...
});
await Desktop.agentContact.consult({
  ...
});
await Desktop.agentContact.decline({
  ...
});
await Desktop.agentContact.consultTransfer({
  ...
});
await Desktop.agentContact.vteamList({
  ...
});
await Desktop.agentContact.pauseRecording({
  ...
});
await Desktop.agentContact.resumeRecording({
  ...
});

// List of available agent-contact aqm notifs events:
Desktop.agentContact.addEventListener("eAgentContact", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentContactAssigned", msg => console.log(msg));

Desktop.agentContact.addEventListener("eAgentContactAssignFailed", msg =>
console.log(msg));
Desktop.agentContact.addEventListener("eAgentContactEnded", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentContactWrappedUp", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentOfferContact", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentOfferContactRona", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentOfferConsult", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentWrapup", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentContactHeld", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentContactUnHeld", msg => console.log(msg));
Desktop.agentContact.addEventListener("eCallRecordingStarted", msg => console.log(msg));
Desktop.agentContact.addEventListener("eResumeRecording", msg => console.log(msg));
Desktop.agentContact.addEventListener("ePauseRecording", msg => console.log(msg));
Desktop.agentContact.addEventListener("eConsultTransfer", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentblindTransferred", msg => console.log(msg));

Desktop.agentContact.addEventListener("eAgentvteamTransfer", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultCreated", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultConferenced", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultEnded", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentCtqCancelled", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultConferenceEnded", msg =>
console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsulting", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultFailed", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultEndFailed", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentCtqFailed", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentCtqCancelFailed", msg => console.log(msg));
Desktop.agentContact.addEventListener("eAgentConsultConferenceEndFailed", msg =>
console.log(msg));

// Module supports removing added listeners like:
const listener = msg => console.log(msg);
Desktop.agentContact.addEventListener("eAgentContact", listener);
Desktop.agentContact.removeEventListener("eAgentContact", listener);

// Module supports one-time added listeners like:
Desktop.agentContact.addOnceEventListener("eAgentContact", listener);

```

```
Desktop.agentContact.removeOnceEventListener("eAgentContact", listener);

// Module supports removing all listeners like:
Desktop.agentContact.removeAllEventListeners();
```

### 応答の例

```
// Generic Response Example form of data is :
data: {
  mediaResourceId: string;
  eventType: string;
  agentId: string;
  destAgentId: string;
  trackingId: string;
  consultMediaResourceId: string;
  interaction: {
    isFcManaged: boolean;
    isTerminated: boolean;
    mediaType: "email" | "chat" | "telephony" | "social" | "sms" | "facebook" |
string;
    previousVTeams: string[];
    state: string;
    currentVTeam: string;
    participants: any; // todo
    interactionId: string;
    orgId: string;
  },
  interactionId: string;
  orgId: string;
  owner: string;
  queueMgr: string;
  type: string;
  ronaTimeout ? : number;
  isConsulted ? : boolean;
  isConferencing: boolean;
  updatedBy ? : string;
  destinationType ? : string;
}
```

- [メソッド \(55 ページ\)](#)
- [非同期イベント \(111 ページ\)](#)

## メソッド

### accept()

着信タスクを受け付けます。

#### 例

```
await Desktop.agentContact.accept({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
});
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい

### 戻り値

{Object} タスクに対応する値。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentContactAssigned、そうでない場合は AgentContactAssignFailed です。

### 応答の例

```
const acceptResponse = {
  "data": {
    "agentId": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "value": "janedoe@gmail.com"
        },
        "category": {
          "value": "Sales"
        },
        "customerName": {
          "value": "Jane Doe"
        },
        "dn": {
          "value": "Desktop"
        },
        "entryPointId": {
          "value": "AXCqFyz1G2pKap9PI-mW"
        },
        "guestId": {
          "value":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MTVmZGIwMTA"
        },
        "mediaChannel": {
          "value": "web"
        },
        "reason": {
          "value": "Test"
        },
        "reasonCode": {
          "value": "Sales"
        },
        "ronaTimeout": {
          "value": "30"
        },
        "roomTitle": {
          "value": "Help Jane Doe with Sales"
        },
        "taskToBeSelfServiced": {
          "value": "false"
        }
      }
    }
  }
}
```



```

        "templateId": {
          "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
        },
        "templateName": {
          "value": "Desktop"
        },
        "virtualTeamName": {
          "value": "Chat_Queue"
        }
      },
      "callAssociatedDetails": {
        "ani": "janedoe@gmail.com",
        "category": "Sales",
        "customerName": "Jane Doe",
        "dn": "Desktop",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MTVmZGIwMTA",
        "mediaChannel": "web",
        "reason": "Test",
        "reasonCode": "Sales",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "virtualTeamName": "Chat_Queue"
      },
      "callFlowParams": {
        "Automation": {
          "description": "(vteam, A valid VTeam.)",
          "name": "Automation",
          "qualifier": "vteam",
          "value": "3270",
          "valueDataType": "string"
        },
        "Debit": {
          "description": "(vteam, A valid VTeam.)",
          "name": "Debit",
          "qualifier": "vteam",
          "value": "3270",
          "valueDataType": "string"
        },
        "Sales": {
          "description": "(vteam, A valid VTeam.)",
          "name": "Sales",
          "qualifier": "vteam",
          "value": "3270",
          "valueDataType": "string"
        }
      },
      "callProcessingDetails": {
        "QMgrName": "aqm",
        "QueueId": "3270",
        "ani": "janedoe@gmail.com",
        "category": "Sales",
        "customerName": "Jane Doe",
        "dnis": "Desktop",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MTVmZGIwMTA",
        "mediaChannel": "web",
        "reason": "Test",
        "reasonCode": "Sales",

```

accept()

```

        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "tenantId": "133",
        "virtualTeamName": "Chat_Queue",
        "vteamId": "AXCqFyz1G2pKap9FI-mW"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "3270",
    "interactionId": "59dbbca6-4194-11eb-881c-253923a967e0",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
        "Y21zY29zcGFyazovL3VzL1JPT00vNjVjOWVlYjAtNDE5NC0xMWViLWE1N2QtZWY2N2MwZTMzYmFh": {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "mainCall",
            "mediaMgr": "cmm",
            "mediaResourceId":
        "Y21zY29zcGFyazovL3VzL1JPT00vNjVjOWVlYjAtNDE5NC0xMWViLWE1N2QtZWY2N2MwZTMzYmFh",
            "mediaType": "chat",
            "participants": ["janedoe@gmail.com",
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed"]
        }
    },
    "mediaChannel": "web",
    "mediaType": "chat",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "participants": {
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed": {
            "channelId": "bff9350a-9d2a-489a-a8d5-d9dacea4b692",
            "consultState": null,
            "consultTimestamp": null,
            "dn": "9997770110",
            "hasJoined": true,
            "id": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
            "isConsulted": false,
            "isWrapUp": false,
            "joinTimestamp": 1608339210824,
            "lastUpdated": 1608339210824,
            "name": "uui-agent4 uui-agent4",
            "pType": "Agent",
            "queueId": "3270",
            "queueMgrId": "aqm",
            "sessionId": "102d2096-bcc0-45bb-a583-a02f6c188071",
            "siteId": "473",
            "teamId": "1940",
            "teamName": "Medical Help",
            "type": "Agent",
            "wrapUpTimestamp": null
        },
        "janedoe@gmail.com": {
            "id": "janedoe@gmail.com",
            "pType": "Customer",
            "type": "Customer"
        }
    }
},

```

```

        "previousVTeams": [],
        "state": "connected"
    },
    "interactionId": "59dbbca6-4194-11eb-881c-253923a967e0",
    "mediaResourceId":
    "Y21zY29zcGFyazovL3VzL1JPT00vNjVjOWV1YjAtNDE5NC0xMWViLWE1N2QtZWY2N2MwZTMwYmFh",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "queueMgr": "aqm",
    "trackingId": "9c9b97e0-4194-11eb-b819-7b8dd50ee0cd",
    "type": "AgentContactAssigned"
  },
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "trackingId": "notifs_02db68c8-a5db-4028-95bb-1aebd6f87609",
  "type": "RoutingMessage"
}

```

## consultAccept()

打診タスクを受け付けます。

### 例

```

await Desktop.agentContact.consultAccept({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
});

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい

### 戻り値

{Object} タスクに対応する値。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentConsulting`、そうでない場合は `AgentContactAssignFailed` です。

### 応答の例

```

const consultAcceptResponse = {
  "data": {
    "agentId": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
    "consultMediaResourceId": "c1b507d0-9aac-4500-a9f8-50f30eba4310",
    "destAgentId": "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "": {
          "agentEditable": false,
          "displayName": "",
          "name": "",
          "type": "STRING",

```

```

        "value": ""
    },
    "IvrPath": {
        "agentEditable": false,
        "displayName": "IvrPath",
        "name": "IvrPath",
        "type": "STRING",
        "value": " EOI"
    },
    "ani": {
        "agentEditable": false,
        "displayName": "ani",
        "name": "ani",
        "type": "STRING",
        "value": "*****"
    },
    "dn": {
        "agentEditable": false,
        "displayName": "dn",
        "name": "dn",
        "type": "STRING",
        "value": "*****"
    },
    "nextVteamId": {
        "agentEditable": false,
        "displayName": "nextVteamId",
        "name": "nextVteamId",
        "type": "STRING",
        "value": "Queue"
    },
    "pathId": {
        "agentEditable": false,
        "displayName": "pathId",
        "name": "pathId",
        "type": "STRING",
        "value": " StartCall PlayDone"
    },
    "ronaTimeout": {
        "agentEditable": false,
        "displayName": "ronaTimeout",
        "name": "ronaTimeout",
        "type": "STRING",
        "value": "30"
    },
    "virtualTeamName": {
        "agentEditable": false,
        "displayName": "virtualTeamName",
        "name": "virtualTeamName",
        "type": "STRING",
        "value": "Queue-1"
    }
},
"callAssociatedDetails": {
    "": "",
    "IvrPath": " EOI",
    "ani": "*****",
    "dn": "*****",
    "nextVteamId": "Queue",
    "pathId": " StartCall PlayDone",
    "ronaTimeout": "30",
    "virtualTeamName": "Queue-1"
},
"callFlowParams": {
    "Queue": {

```

```

        "description": "(vteam, A valid VTeam.)",
        "name": "Queue",
        "qualifier": "vteam",
        "value": "1336",
        "valueDataType": "string"
    },
    "WelcomePrompt": {
        "description": "(mediaFile, A valid media file.)",
        "name": "WelcomePrompt",
        "qualifier": "mediaFile",
        "value": "GAwelcome.wav",
        "valueDataType": "string"
    }
},
"callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "1336",
    "ani": "*****",
    "dnis": "*****",
    "isPaused": "false",
    "jscriptId": "AW7HP3QNB44q0sL-cTVz",
    "pauseDuration": "30",
    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "82",
    "virtualTeamName": "Queue-1",
    "vteamId": "1336"
},
"contactDirection": {
    "type": "INBOUND"
},
"currentVTeam": "1336",
"interactionId": "1277da79-dc49-439c-ab72-f00453a1ff71",
"isFcManaged": false,
"isTerminated": false,
"media": {
    "1277da79-dc49-439c-ab72-f00453a1ff71": {
        "holdTimestamp": 1612429578817,
        "isHold": true,
        "mType": "mainCall",
        "mediaMgr": "vmm",
        "mediaResourceId": "1277da79-dc49-439c-ab72-f00453a1ff71",
        "mediaType": "telephony",
        "participants": [
            "*****",
            "9b036d89-930c-4187-a5bd-5dbb1439fe41"
        ]
    }
},
"clb507d0-9aac-4500-a9f8-50f30eba4310": {
    "holdTimestamp": null,
    "isHold": false,
    "mType": "consult",
    "mediaMgr": "vmm",
    "mediaResourceId": "clb507d0-9aac-4500-a9f8-50f30eba4310",
    "mediaType": "telephony",
    "participants": [
        "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f",
        "9b036d89-930c-4187-a5bd-5dbb1439fe41",
        "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f"
    ]
}
},

```

```

"mediaChannel": "broadcloud",
"mediaType": "telephony",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"outboundType": null,
"owner": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
"participants": {
  "*****": {
    "id": "*****",
    "pType": "Customer",
    "type": "Customer"
  },
  "9b036d89-930c-4187-a5bd-5dbb1439fe41": {
    "channelId": "3de6706f-95b7-485a-9304-30928f7ee52e",
    "consultState": "consulting",
    "consultTimestamp": 1612429579220,
    "dn": "*****",
    "hasJoined": true,
    "id": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
    "isConsulted": true,
    "isWrapUp": false,
    "joinTimestamp": 1612429549032,
    "lastUpdated": 1612429549032,
    "name": "user1",
    "pType": "Agent",
    "queueId": "1336",
    "queueMgrId": "aqm",
    "sessionId": "90896c91-0f77-4bc6-aec3-488a34ca88d3",
    "siteId": "205",
    "teamId": "598",
    "teamName": "integ-test-team",
    "type": "Agent",
    "wrapUpTimestamp": null
  },
  "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f": {
    "channelId": "fb704337-71c3-4802-a03b-84a2d18eeaa0",
    "consultState": "consulting",
    "consultTimestamp": 1612429579220,
    "dn": "*****",
    "hasJoined": true,
    "id": "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f",
    "isConsulted": true,
    "isWrapUp": false,
    "joinTimestamp": 1612429579200,
    "lastUpdated": 1612429579200,
    "name": "user2",
    "pType": "Agent",
    "queueId": "1336",
    "queueMgrId": "aqm",
    "sessionId": "57323f8b-9460-45fa-aaf6-4ca51c327af8",
    "siteId": "205",
    "teamId": "598",
    "teamName": "integ-test-team",
    "type": "Agent",
    "wrapUpTimestamp": null
  }
},
"previousVTeams": [
  "AW6h1AVRrZSGni185x28"
],
"state": "consulting",
"workflowManager": null
},
"interactionId": "1277da79-dc49-439c-ab72-f00453alff71",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",

```

```

    "queueMgr": "aqm",
    "trackingId": "e7c032be-b18e-4b63-8ab2-6baf50b137b1",
    "type": "AgentConsulting"
  },
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "trackingId": "notifs_3464be63-83e7-4b2d-94d5-a63fddc108d5",
  "type": "RoutingMessage"
}

```

## end()

特定のタスクのインタラクションを終了します。

### 例

```

await Desktop.agentContact.end({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
});

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい

### 戻り値

{Object} タスクに対応する値。



(注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentWrapup` または `ContactEnded`、そうでない場合は `ContactEndFailed` です。

### 応答の例

```

const endResponse = {
  "data": {
    "agentsPendingWrapUp": ["7d12d9ea-e8e0-41ee-81bf-c11a685b64ed"],
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "value": "janedoe@gmail.com"
        },
        "category": {
          "value": "Sales"
        },
        "customerName": {
          "value": "Jane Doe"
        },
        "dn": {
          "value": "Desktop"
        },
        "entryPointId": {
          "value": "AXCqFyz1G2pKap9PI-mW"
        }
      }
    }
  }
}

```

end()

```

      "guestId": {
        "value":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1M1kZjM3LTRlODEtOGMxM1mZjkyMTE3MmQ1NzY"
      },
      "mediaChannel": {
        "value": "web"
      },
      "reason": {
        "value": "Test"
      },
      "reasonCode": {
        "value": "Sales"
      },
      "ronaTimeout": {
        "value": "30"
      },
      "roomTitle": {
        "value": "Help Jane Doe with Sales"
      },
      "taskToBeSelfServiced": {
        "value": "false"
      },
      "templateId": {
        "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
      },
      "templateName": {
        "value": "Desktop"
      },
      "virtualTeamName": {
        "value": "Chat_Queue"
      }
    },
    "callAssociatedDetails": {
      "ani": "janedoe@gmail.com",
      "category": "Sales",
      "customerName": "Jane Doe",
      "dn": "Desktop",
      "entryPointId": "AXCqFyz1G2pKap9PI-mW",
      "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1M1kZjM3LTRlODEtOGMxM1mZjkyMTE3MmQ1NzY",
      "mediaChannel": "web",
      "reason": "Please help",
      "reasonCode": "Sales",
      "ronaTimeout": "30",
      "roomTitle": "Help Jane Doe with Sales",
      "taskToBeSelfServiced": "false",
      "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
      "templateName": "Desktop",
      "virtualTeamName": "Chat_Queue"
    },
    "callFlowParams": {
      "Automation": {
        "description": "(vteam, A valid VTeam.)",
        "name": "Automation",
        "qualifier": "vteam",
        "value": "3270",
        "valueDataType": "string"
      },
      "Debit": {
        "description": "(vteam, A valid VTeam.)",
        "name": "Debit",
        "qualifier": "vteam",
        "value": "3270",
        "valueDataType": "string"
      }
    }
  }
}

```



```

    },
    "Sales": {
      "description": "(vteam, A valid VTeam.)",
      "name": "Sales",
      "qualifier": "vteam",
      "value": "3270",
      "valueDataType": "string"
    }
  },
  "callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3270",
    "ani": "janedoe@gmail.com",
    "category": "Sales",
    "customerName": "Jane Doe",
    "dnis": "Desktop",
    "entryPointId": "AXCqFyz1G2pKap9PI-mW",
    "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTR1ODEtOGMxMilmZjkyMTE3MmQ1NzY",
    "mediaChannel": "web",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "reason": "Please help",
    "reasonCode": "Sales",
    "ronaTimeout": "30",
    "roomTitle": "Help Jane Doe with Sales",
    "taskToBeSelfServiced": "false",
    "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
    "templateName": "Desktop",
    "tenantId": "133",
    "virtualTeamName": "Chat_Queue",
    "vteamId": "AXCqFyz1G2pKap9PI-mW"
  },
  "contactDirection": {
    "type": "INBOUND"
  },
  "currentVTeam": "3270",
  "interactionId": "a66d05ab-40ab-11eb-b59a-7d3326b3ffc1",
  "isFcManaged": false,
  "isTerminated": true,
  "media": {},
  "mediaChannel": "web",
  "mediaType": "chat",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": null,
  "owner": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
  "participants": {
    "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed": {
      "channelId": "ff0a5296-8aa2-483b-8c50-577fedb1bfde",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "9997770110",
      "hasJoined": true,
      "id": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1608239195992,
      "lastUpdated": 1608239195992,
      "name": "uui-agent4 uui-agent4",
      "pType": "Agent",
      "queueId": "3270",
      "queueMgrId": "aqm",
      "sessionId": "b8c096ab-f9a0-49be-9efd-f8c78e1a7061",
      "siteId": "473",

```

```

        "teamId": "962",
        "teamName": "Team X",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "janedoe@gmail.com": {
        "id": "janedoe@gmail.com",
        "pType": "Customer",
        "type": "Customer"
    }
},
"previousVTeams": [],
"state": "connected"
},
"interactionId": "a66d05ab-40ab-11eb-b59a-7d3326b3ffc1",
"mediaResourceId":
"Y2lzY29zcGFyazovL3VzL1JPT00vYWEzNTA5YzAtNDZhYi0xMWViLWE1N2QtZWY2N2MwZTM5YmFh",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "4eb68870-40ac-11eb-8866-d7ef6ff89aa6",
"type": "ContactEnded"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_3cda2c34-1410-4476-9c5c-5d779a615b8a",
"type": "RoutingMessage"
}
}

```

## buddyAgents()

打診コールおよび会議コールに対応可能なエージェントのリストを取得します。

### 例

```

await Desktop.agentContact.buddyAgents({
  data: {
    agentProfileId: "AXCLfZhH9S1oTdqE1OFw",
    channelName: "chat",
    state: "Available"
  }
});

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
data	オブジェクト	バディエージェントのオプション。	はい
-->agentProfileId	文字列	エージェントプロファイルの一意の識別子。	はい
-->channelName	文字列	チャットなどのメディアチャネルタイプ。	はい
-->state	文字列	エージェントの対応可否ステータス。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `BuddyAgents`、そうでない場合は `BuddyAgentsRetrieveFailed`。

### 応答の例

```
type BuddyAgentResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "agentList": [],
    "agentSessionId": "25f31485-f41c-4cca-abe5-e12b80735715",
    "eventType": "AgentDesktopMessage",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "trackingId": "1a2737c0-6704-11eb-9a89-971a3ca976b4",
    "type": "BuddyAgents"
  },
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "trackingId": "notifs_49119bed-6dbc-48e6-89fa-9dde8da3fc3e",
  "type": "BuddyAgents"
}
```

## consult()

`buddyAgents` のリストから、いずれかのエージェントを使用して打診リクエストを開始します。

### 例

```
await Desktop.agentContact.consult({
  interactionId: "a26d035d - b547 - 11 ea - 984 b - d3128111ce6b",
  data: {
    agentId: "43 bfcdef - 5551 - 45 b1 - a8f9 - a0b33e66e3fe",
    destAgentId: "f795f41f - 3782 - 44 fa - 97 a4 - c8b4dc029477",
    mediaType: "chat"
  },
  url: "consult"
});
// consult Payload Type
type ConsultPayload = {
  interactionId: string;
  data: ConsultPayload | ConsultDN | ConsultAgent | ConsultQueue;
  url: string;
}

type ConsultPayload = {
  agentId: string;
  destAgentId: string | undefined;
  mediaType ? : string | undefined;
}

type ConsultDN = {
  destAgentId: string;
  destinationType: string;
  mediaType: string;
  trackingId ? : string;
};

type ConsultAgent = {
  agentId: string;
```

```

    destAgentId: string;
    destAgentDN ? : string;
    destAgentTeamId ? : string;
    destSiteId ? : string;
    mediaType: string;
    trackingId: string;
  };

  type ConsultQueue = {
    agentId: string;
    queueId: string;
    trackingId: string;
  };

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	打診リクエストのオプション。	はい
-->agentId	文字列	エージェントの一意の識別子。	はい
-->destAgentId	文字列	打診リクエストの接続先であるエージェントの一意の識別子。	はい
-->mediaType	文字列	チャットなどのメディアチャンネルタイプ。	はい
-->destinationType	文字列	送信する打診リクエストのタイプ。接続先タイプには、エージェント、キュー、または DN を使用できます。	はい
-->trackingId	文字列	打診リクエストタイプに基づく一意の識別子。	はい
-->destAgentDN	文字列	接続先エージェントのダイヤル番号。	はい
-->destAgentTeamId	文字列	接続先エージェントが属するチームの一意の識別子。	はい
-->destSiteId	文字列	接続先サイトの一意の識別子。	はい
-->queueId	文字列	キューの一意の識別子。	はい
url	文字列	打診の URL。例 : consult, ctq	はい

## 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentConsultCreated`、そうでない場合は `AgentCtqFailed` または `AgentConsultFailed` です。

### 応答の例

```
const consultResponse = {
  "data": {
    "agentId": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "consultMediaResourceId":
"consult_Y21zY29zcGFyazovL3VzL1JPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGItyjkyYTZkMmI5NTI1",
    "destAgentId": "aad323de-32d8-48c9-af6b-b68dfbabfe20",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "value": "janedoe@gmail.com"
        },
        "category": {
          "value": "Sales"
        },
        "customerName": {
          "value": "Jane Doe"
        },
        "dn": {
          "value": "Desktop"
        },
        "entryPointId": {
          "value": "AXCqFyz1G2pKap9PI-mW"
        },
        "guestId": {
          "value":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTRlODEtOGMxMilmZjkyMTE3MmQ1NzY"
        },
        "mediaChannel": {
          "value": "web"
        },
        "reason": {
          "value": "test"
        },
        "reasonCode": {
          "value": "Sales"
        },
        "ronaTimeout": {
          "value": "30"
        },
        "roomTitle": {
          "value": "Help Jane Doe with Sales"
        },
        "taskToBeSelfServiced": {
          "value": "false"
        },
        "templateId": {
          "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
        },
        "templateName": {
          "value": "Desktop"
        },
        "virtualTeamName": {
```

```

        "agentEditable": false,
        "displayName": "virtualTeamName",
        "name": "virtualTeamName",
        "type": "STRING",
        "value": "Chat_Queue"
    }
},
"callAssociatedDetails": {
    "ani": "janedoe@gmail.com",
    "category": "Sales",
    "customerName": "Jane Doe",
    "dn": "Desktop",
    "entryPointId": "AXCqFyz1G2pKap9PI-mW",
    "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTRlODEtOGMxMilmZjkyMTE3MmQ1NzY",
    "mediaChannel": "web",
    "reason": "test",
    "reasonCode": "Sales",
    "ronaTimeout": "30",
    "roomTitle": "Help Jane Doe with Sales",
    "taskToBeSelfServiced": "false",
    "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
    "templateName": "Desktop",
    "virtualTeamName": "Chat_Queue"
},
"callFlowParams": {
    "Automation": {
        "description": "(vteam, A valid VTeam.)",
        "name": "Automation",
        "qualifier": "vteam",
        "value": "3270",
        "valueDataType": "string"
    },
    "Debit": {
        "description": "(vteam, A valid VTeam.)",
        "name": "Debit",
        "qualifier": "vteam",
        "value": "3270",
        "valueDataType": "string"
    },
    "Sales": {
        "description": "(vteam, A valid VTeam.)",
        "name": "Sales",
        "qualifier": "vteam",
        "value": "3270",
        "valueDataType": "string"
    }
},
"callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3270",
    "ani": "janedoe@gmail.com",
    "category": "Sales",
    "customerName": "Jane Doe",
    "dnis": "Desktop",
    "entryPointId": "AXCqFyz1G2pKap9PI-mW",
    "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTRlODEtOGMxMilmZjkyMTE3MmQ1NzY",
    "mediaChannel": "web",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "reason": "test",
    "reasonCode": "Sales",
    "ronaTimeout": "30",

```

```

        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "tenantId": "133",
        "virtualTeamName": "Chat_Queue",
        "vteamId": "AXCqFyz1G2pKap9PI-mW"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "3270",
    "interactionId": "5622d599-40af-11eb-8606-657fd6cc0548",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
        "Y21zY29zcGFyazovL3VzL1JPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGItYjkyYTZkMmI5NTI1": {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "mainCall",
            "mediaMgr": "cmm",
            "mediaResourceId":
        "Y21zY29zcGFyazovL3VzL1JPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGItYjkyYTZkMmI5NTI1",
            "mediaType": "chat",
            "participants": ["janedoe@gmail.com",
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed"]
        },
        "consult_Y21zY29zcGFyazovL3VzL1JPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGItYjkyYTZkMmI5NTI1":
        {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "consult",
            "mediaMgr": "cmm",
            "mediaResourceId":
        "consult_Y21zY29zcGFyazovL3VzL1JPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGItYjkyYTZkMmI5NTI1",
            "mediaType": "chat",
            "participants": ["aad323de-32d8-48c9-af6b-b68dfbabfe20",
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed"]
        }
    },
    "mediaChannel": "web",
    "mediaType": "chat",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "participants": {
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed": {
            "channelId": "7272c119-821f-485a-b352-29belf3f0fff",
            "consultState": "consultInitiated",
            "consultTimestamp": 1608240860457,
            "dn": "9997770110",
            "hasJoined": true,
            "id": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
            "isConsulted": false,
            "isWrapUp": false,
            "joinTimestamp": 1608240840633,
            "lastUpdated": 1608240840633,
            "name": "uui-agent4 uui-agent4",
            "pType": "Agent",
            "queueId": "3270",
            "queueMgrId": "aqm",
            "sessionId": "b8c096ab-f9a0-49be-9efd-f8c78e1a7061",

```

```

        "siteId": "473",
        "teamId": "962",
        "teamName": "Team X",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "aad323de-32d8-48c9-af6b-b68dfbabfe20": {
        "channelId": "b1da7e42-2aeb-492b-ad06-115ba8dfa506",
        "consultState": null,
        "consultTimestamp": null,
        "dn": "9997770109",
        "hasJoined": false,
        "id": "aad323de-32d8-48c9-af6b-b68dfbabfe20",
        "isConsulted": true,
        "isWrapUp": false,
        "joinTimestamp": null,
        "lastUpdated": 1608240860436,
        "name": "uui-agent2 uui-agent2",
        "pType": "Agent",
        "queueId": "3270",
        "queueMgrId": "aqm",
        "sessionId": "3443f9d7-68de-42a0-a590-176a37a3565a",
        "siteId": "473",
        "teamId": "962",
        "teamName": "Team X",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "janedoe@gmail.com": {
        "id": "janedoe@gmail.com",
        "pType": "Customer",
        "type": "Customer"
    }
},
"previousVTeams": [],
"state": "consult"
},
"interactionId": "5622d599-40af-11eb-8606-657fd6cc0548",
"mediaResourceId":
"Y21zY29zcGFyYXovL3VzLlJPT00vNjJkYjBmMzAtNDBhZi0xMWViLWE3ZGIYkyYTZkMmI5NTI1",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "9f682870-40af-11eb-bc9c-df3ff755538c",
"type": "AgentConsultCreated"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_6dc77353-e810-4751-82ac-dab03e3951c2",
"type": "RoutingMessage"
}

```

## consultConference()

buddyAgents のリストから、いずれかのエージェントを使用して会議リクエストを開始します。

### 例

```

await Desktop.agentContact.consultConference({
  interactionId: "a26d035d - b547 - 11 ea - 984 b - d3128111ce6b",
  data: {
    agentId: "43 bfcdef - 5551 - 45 b1 - a8f9 - a0b33e66e3fe",
    destAgentId: "f795f41f - 3782 - 44 fa - 97 a4 - c8b4dc029477",
    mediaType: "chat"
  }
})

```



```

});
// consult conference Payload Type

type ConsultConferencePayload = {
  interactionId: string; data: ConsultPayload | ConsultDN
}

type ConsultDN = {
  destAgentId: string;
  destinationType: string;
  mediaType: string;
  trackingId ? : string | undefined;
}

type ConsultPayload = {
  agentId: string;
  destAgentId: string | undefined;
  mediaType ? : string | undefined;
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	会議リクエストのオプション。	はい
-->agentId	文字列	エージェントの一意の識別子。	はい
-->destAgentId	文字列	会議の接続先であるエージェントの一意の識別子。	はい
-->mediaType	文字列	チャットなどのメディアチャネルタイプ。	はい
-->destinationType	文字列	開催するコンサルト会議のタイプ。接続先タイプには、エージェント、キュー、またはDNを使用できます。	はい
-->trackingId	文字列	会議タイプに基づく一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentConsultConferenced、そうでない場合は AgentConsultConferenceFailed です。

### 応答の例

```

const consultConferenceResponse = {
  "data": {
    "agentId": "337c1cca-d79b-44fc-8d8e-20dlb5e4dadf",
    "consultMediaResourceId":
"consult_Y2lzY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMmVhLTk0YjAtNzdjZDBkYWVjNDA1",
    "destAgentId": "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "agentEditable": false,
          "displayName": "ani",
          "name": "ani",
          "type": "STRING",
          "value": "displayName"
        },
        "appUser": {
          "agentEditable": false,
          "displayName": "appUser",
          "name": "appUser",
          "type": "STRING",
          "value": "appuser-1212"
        },
        "customerName": {
          "agentEditable": false,
          "displayName": "customerName",
          "name": "customerName",
          "type": "STRING",
          "value": "displayName"
        },
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "AXVP2lxFOG6K4V3pFKRG"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        },
        "virtualTeamName": {
          "agentEditable": false,
          "displayName": "virtualTeamName",
          "name": "virtualTeamName",
          "type": "STRING",
          "value": "QA-e2e-SocialQueue"
        }
      }
    },
    "callAssociatedDetails": {
      "ani": "*****",
      "appUser": "appuser-1212",
      "customerName": "displayName",
      "dn": "*****",
      "ronaTimeout": "30",
      "virtualTeamName": "QA-e2e-SocialQueue"
    },
    "callFlowParams": {},
    "callProcessingDetails": {
      "QMgrName": "aqm",

```

```

        "QueueId": "5134",
        "ani": "*****",
        "appUser": "appuser-1212",
        "customerName": "displayName",
        "dnis": "*****",
        "pauseDuration": "30",
        "pauseResumeEnabled": "true",
        "ronaTimeout": "30",
        "taskToBeSelfServiced": "false",
        "tenantId": "82",
        "virtualTeamName": "QA-e2e-SocialQueue",
        "vteamId": "5134"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "5134",
    "interactionId": "1da228dc-66cc-11eb-b852-ed71847d1aec",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
        "Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1": {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "mainCall",
            "mediaMgr": "socialmm",
            "mediaResourceId":
        "Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1",
            "mediaType": "social",
            "participants": [
                "displayName",
                "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
                "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2"
            ]
        },
        "consult_Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1": {
            {
                "holdTimestamp": null,
                "isHold": false,
                "mType": "consult",
                "mediaMgr": "socialmm",
                "mediaResourceId":
        "consult_Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1",
                "mediaType": "social",
                "participants": [
                    "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2",
                    "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
                    "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2"
                ]
            }
        },
        "mediaChannel": "type",
        "mediaType": "social",
        "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
        "outboundType": null,
        "owner": "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
        "participants": {
            "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf": {
                "channelId": "e08a6068-688d-472b-887b-1c5d39c7155f",
                "consultState": "conferencing",
                "consultTimestamp": 1612431260416,
                "dn": "*****",

```

```

        "hasJoined": true,
        "id": "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
        "isConsulted": false,
        "isWrapUp": false,
        "joinTimestamp": 1612431253790,
        "lastUpdated": 1612431253790,
        "name": "social-agent2",
        "pType": "Agent",
        "queueId": "5134",
        "queueMgrId": "aqm",
        "sessionId": "a9aa9867-e62f-4415-9ea4-6babace39d4f",
        "siteId": "205",
        "teamId": "384",
        "teamName": "TeamSales",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2": {
        "channelId": "841235e4-7052-455b-be64-30f2f3c4a59b",
        "consultState": "conferencing",
        "consultTimestamp": 1612431260416,
        "dn": "*****",
        "hasJoined": true,
        "id": "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2",
        "isConsulted": true,
        "isWrapUp": false,
        "joinTimestamp": 1612431258137,
        "lastUpdated": 1612431258137,
        "name": "social-agent1",
        "pType": "Agent",
        "queueId": "5134",
        "queueMgrId": "aqm",
        "sessionId": "600e7ebb-20aa-4c91-a4f0-ebc596249603",
        "siteId": "205",
        "teamId": "384",
        "teamName": "TeamSales",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "displayName": {
        "id": "displayName",
        "pType": "Customer",
        "type": "Customer"
    }
},
"previousVTeams": [
    "AXVP21xF0G6K4V3pFKRG"
],
"state": "conference",
"workflowManager": null
},
"interactionId": "1da228dc-66cc-11eb-b852-ed71847d1aec",
"mediaResourceId":
"Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "4a958ace-49a2-49aa-b2e5-8798ede03f13",
"type": "AgentConsultConferenced"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_68c519c8-04f5-48b4-a5cf-609a09ccd765",
"type": "RoutingMessage"
}

```

## consultEnd()

打診リクエストを終了します。

### 例

```
await Desktop.agentContact.consultEnd({
  interactionId: "a26d035d - b547 - 11 ea - 984 b - d3128111ce6b",
  isConsult: false
});

// Consult end payload type
{
  interactionId: string;isConsult: boolean
}
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
isConsult	ブール値	タスクリクエストが打診かどうかを決定します。 <ul style="list-style-type: none"> <li>• True : タスクリクエストは打診です。</li> <li>• False : タスクリクエストは打診ではありません。</li> </ul>	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



(注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功し、`isConsult` が **true** である場合は、値は `AgentConsultEnded`、そうでない場合は `AgentConsultConferenceEnded` です。

メソッドが失敗した場合、値は `AgentConsultEndFailed` です。

### 応答の例

```
const consultEndResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "consultMediaResourceId": "f23b5b8a-4b02-451e-b9f5-8517b5bef97c",
    "destAgentId": "9fca6158-a8bd-43e6-aeba-e5055aea08c5",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",

```

```

        "value": "8895579172"
    },
    "ronaTimeout": {
        "agentEditable": false,
        "displayName": "ronaTimeout",
        "name": "ronaTimeout",
        "type": "STRING",
        "value": "30"
    },
    "virtualTeamName": {
        "agentEditable": false,
        "displayName": "virtualTeamName",
        "name": "virtualTeamName",
        "type": "STRING",
        "value": "Outdial Queue-1"
    }
},
"callAssociatedDetails": {
    "dn": "8895579172",
    "ronaTimeout": "30",
    "virtualTeamName": "Outdial Queue-1"
},
"callFlowParams": {
    "OutdialQueue": {
        "description": "(vteam, The Outdial Queue.)",
        "name": "OutdialQueue",
        "qualifier": "vteam",
        "value": "3264",
        "valueDataType": "string"
    }
},
"callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3264",
    "dnis": "8895579172",
    "isPaused": "false",
    "outdialTransferToQueueEnabled": "false",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "virtualTeamName": "Outdial Queue-1",
    "vteamId": "AXCLfZZU9S1oTdqe1OFZ"
},
"contactDirection": {
    "type": "OUTBOUND"
},
"currentVTeam": "3264",
"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"isFcManaged": false,
"isTerminated": false,
"media": {
    "1dd08726-bca5-4ce8-9b51-b20dca8ab154": {
        "holdTimestamp": 1612356600241,
        "isHold": true,
        "mType": "mainCall",
        "mediaMgr": "vmm",
        "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
        "mediaType": "telephony",
        "participants": [
            ""
        ]
    }
}

```

```

    }
  },
  "mediaChannel": "dialer",
  "mediaType": "telephony",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": "OUTDIAL",
  "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
  "participants": {
    "": {
      "id": "",
      "pType": "Customer",
      "type": "Customer"
    },
    "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
      "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "8895579172",
      "hasJoined": true,
      "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1612354635705,
      "lastUpdated": 1612354635705,
      "name": "John Doe",
      "pType": "Agent",
      "queueId": "3264",
      "queueMgrId": "aqm",
      "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
      "siteId": "472",
      "teamId": "960",
      "teamName": "Email_Team",
      "type": "Agent",
      "wrapUpTimestamp": null
    }
  },
  "previousVTeams": [
  ],
  "state": "connected",
  "workflowManager": null
},
"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "f7c7ecd2-ef27-4273-bc11-1543bddc8810",
"type": "AgentConsultEnded"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_860dle16-88b8-4d6c-8bac-18075d72a7e8",
"type": "RoutingMessage"
}
}

```

## decline()

打診リクエストを却下します。

### 例

```

await Desktop.agentContact.decline({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3";
  data: {
    mediaResourceId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
  }
});

```

```

    }
    isConsult: true;
  });

  // Decline payload type
  {
    interactionId: string;
    data: Contact.declinePayload;
    isConsult: boolean
  }

  type declinePayload = {
    mediaResourceId: string;
  }

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	打診リクエストのオプション。	はい
-->mediaResourceId	文字列	メディアリソースの一意の識別子。	はい
isConsult	ブール	タスクリクエストが打診かどうかを決定します。  <ul style="list-style-type: none"> <li>• True : タスクリクエストは打診です。</li> <li>• False : タスクリクエストは打診ではありません。</li> </ul>	はい

## 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功し、isConsult が true である場合は、値は AgentConsultFailed、そうでない場合は AgentOfferContactRona です。

## 応答の例

```

const declineResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "destAgentId": "f795f41f-3782-44fa-97a4-c8b4dc029477",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {},
      "callAssociatedDetails": {},
      "callFlowParams": {},

```



```

    "callProcessingDetails": {},
    "contactDirection": {
      "type": "INBOUND"
    },
    "currentVTeam": "",
    "interactionId": "9ad96648-415b-11eb-881c-ed6dd0d45a04",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {},
    "mediaChannel": "none",
    "mediaType": "none",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": null,
    "participants": {},
    "previousVTeams": [],
    "state": "none",
    "workflowManager": null
  },
  "interactionId": "a26d035d-b547-11ea-984b-d3128111ce6b",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "queueMgr": "aqm",
  "reason": "Test",
  "reasonCode": 500,
  "trackingId": "2037cb70-653d-11eb-8a11-a3c35f103d4b",
  "type": "AgentConsultFailed"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_a241b1c4-3d42-4ad0-8738-71b1e7611550",
"type": "RoutingMessage"
}

```

## cancelCtq()

キューリクエストに対する打診を取り消します。

### 例

```

await Desktop.agentContact.cancelCtq({
  interactionId: "a26d035d - b547 - 11 ea - 984 b - d3128111ce6b",
  data: {
    agentId: "f795f41f - 3782 - 44 fa - 97 a4 - c8b4dc029477",
    queueId: "3268"
  }
});

// cancelCtq Payload
{
  interactionId: string; data: Contact.cancelCtq
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	キューリクエストに対する打診のオプション。	はい
-->agentId	文字列	エージェントの一意の識別子。	はい

名前	タイプ	説明	必須
-->queueId	文字列	キューの一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentCtqCancelled、そうでない場合は AgentCtqCancelFailed です。

### 応答の例

```
const cancelCtqResponse = {
  "data": {
    "agentId": "a96ffe92-f5c8-4715-9427-fa32a7e57ccc",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {},
      "callAssociatedDetails": {},
      "callFlowParams": {},
      "callProcessingDetails": {},
      "contactDirection": {
        "type": "INBOUND"
      },
      "currentVTeam": "",
      "interactionId": "9ad96648-415b-11eb-881c-ed6dd0d45a04",
      "isFcManaged": false,
      "isTerminated": false,
      "media": {},
      "mediaChannel": "none",
      "mediaType": "none",
      "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
      "outboundType": null,
      "owner": null,
      "participants": {},
      "previousVTeams": [],
      "state": "none"
    },
    "interactionId": "c8e35eec-a977-409b-922d-7d2825c63cf2",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "queueId": "7352",
    "queueMgr": "aqm",
    "trackingId": "ee2bc7e0-50c8-11eb-a1f8-5d0ad357b8a5",
    "type": "AgentCtqCancelled"
  },
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "trackingId": "notifs_908141bb-3f1f-43fe-b7e2-5a132c97745f",
  "type": "RoutingMessage"
}
```

## wrapup()

タスクの作業をラップアップします。ラップアップ理由は、エージェントまたは顧客がタスクを終了した後に適用されます。ラップアップは必須であり、エージェントによるタスク処理フローの最後のステップです。

### 例

```
await Desktop.agentContact.wrapup({
  interactionId: "dab1a2f0 - bad0 - 11 ea - 8e7 c - 0 d99ede2535a",
  data: {
    wrapUpReason: "Example reason here";
    auxCodeId: "0";
    isAutoWrapup: "false";
  }
});

// Wrap Payload Type

{
  interactionId: string; data: Contact.WrapupPayload
}
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意的識別子。	はい
data	オブジェクト	ラップアップのオプション。	はい
-->wrapUpReason	文字列	顧客がコンタクトセンターにコールした理由を示すラップアップ理由。	はい
-->auxCodeId	文字列	エージェントの状態を表す一意的識別子。	はい
-->isAutoWrapup	ブール値	自動ラップアップが有効かどうかを決定します。 True : 自動ラップアップが有効です。 False : 自動ラップアップは無効です。	

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentWrappedUp、そうでない場合は AgentWrapupFailed です。

### 応答の例

```

const wrapUpResponse = {
  "data": {
    "agentId": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "value": "janedoe@gmail.com"
        },
        "category": {
          "value": "Sales"
        },
        "customerName": {
          "value": "Jane Doe"
        },
        "dn": {
          "value": "Desktop"
        },
        "entryPointId": {
          "value": "AXCqFyz1G2pKap9PI-mW"
        },
        "guestId": {
          "value":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1Ml1kZjM3LTR1ODEtOGMxMl1mZjkyMTE3MmQ1NzY"
        },
        "mediaChannel": {
          "value": "web"
        },
        "reason": {
          "value": "Test"
        },
        "reasonCode": {
          "value": "Sales"
        },
        "ronaTimeout": {
          "value": "30"
        },
        "roomTitle": {
          "value": "Help Jane Doe with Sales"
        },
        "taskToBeSelfServiced": {
          "value": "false"
        },
        "templateId": {
          "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
        },
        "templateName": {
          "value": "Desktop"
        },
        "virtualTeamName": {
          "value": "Chat_Queue"
        }
      },
      "callAssociatedDetails": {
        "ani": "janedoe@gmail.com",
        "category": "Sales",
        "customerName": "Jane Doe",
        "dn": "Desktop",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1Ml1kZjM3LTR1ODEtOGMxMl1mZjkyMTE3MmQ1NzY",
        "mediaChannel": "web",
        "reason": "Test",

```

```

        "reasonCode": "Sales",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "virtualTeamName": "Chat_Queue"
    },
    "callFlowParams": {
        "Automation": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Automation",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Debit": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Debit",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Sales": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Sales",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        }
    },
    "callProcessingDetails": {
        "QMgrName": "aqm",
        "QueueId": "3270",
        "ani": "janedoe@gmail.com",
        "category": "Sales",
        "customerName": "Jane Doe",
        "dnis": "Desktop",
        "entryPointId": "AXCqFyzlG2pKap9PI-mW",
        "guestId":
        "Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTRlODEtOGMxMilmZjkyMTE3MmQ1NzY",
        "mediaChannel": "web",
        "pauseDuration": "10",
        "pauseResumeEnabled": "true",
        "reason": "Test",
        "reasonCode": "Sales",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "tenantId": "133",
        "virtualTeamName": "Chat_Queue",
        "vteamId": "AXCqFyzlG2pKap9PI-mW"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "3270",
    "interactionId": "a66d05ab-40ab-11eb-b59a-7d3326b3ffc1",
    "isFcManaged": false,
    "isTerminated": true,
    "media": {},
    "mediaChannel": "web",

```

```

    "mediaType": "chat",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "participants": {
      "janedoe@gmail.com": {
        "id": "janedoe@gmail.com",
        "pType": "Customer",
        "type": "Customer"
      }
    }
  },
  "previousVTeams": [],
  "state": "connected"
},
"interactionId": "a66d05ab-40ab-11eb-b59a-7d3326b3ffc1",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "5272a610-40ac-11eb-8866-d7ef6ff89aa6",
"type": "AgentWrappedUp",
"wrapUpAuxCodeId": "0"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_b5e0ebb7-fd1d-479f-8663-81a64030cee6",
"type": "RoutingMessage"
}
}

```

## vTeamList()

転送可能な vTeam (キュー) のリストを取得します。インタラクションを特定のエージェントに転送する代わりに、適切なキュー (vTeam) に転送できます。

### 例

```

await Desktop.agentContact.vteamList({
  data: {
    agentProfileId: "AXCLfZhH9S1oTdqE1OFw",
    agentSessionId: "5 a84d32c - 691 b - 4500 - b163 - d6cdba2a3163",
    channelType: "chat",
    type: "inboundqueue"
    trackingId: "9c9b97e0-4194-11eb-b819-7b8dd50ee0cd",
  }
});

// Vteam List Payload Type
{
  data: Contact.VTeam
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
data	オブジェクト	転送可能なキューのリストのオプション。	はい
-->agentProfileId	文字列	エージェントプロファイルの一意の識別子。	はい
-->agentSessionId	文字列	エージェントセッションの一意の識別子。	はい
-->channelType	文字列	チャットなどのメディアチャンネルタイプ。	はい

名前	タイプ	説明	必須
-->type	文字列	キューリストのタイプ。	はい
-->trackingId	文字列	転送可能なキューリストに基づく一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は vteamList、そうでない場合は vteamListFailed です。

### 応答の例

```
const vTeamResponse = {
  "data": {
    "agentSessionId": "5a84d32c-691b-4500-b163-d6cdba2a3163",
    "callData": "",
    "data": {
      "allowConsultToQueue": true,
      "vteamList": [{
        "analyzerId": "AXT3_BAV0G6K4V3p9euK",
        "callDistributionGroups": [],
        "channelType": "chat",
        "id": "7178",
        "maxTimeInQueue": 500,
        "name": "Chat_Demo",
        "routingType": null,
        "skillBasedRoutingType": null,
        "type": "inboundqueue"
      }, {
        "analyzerId": "AXNSD0dSQJBc-emowhp3",
        "callDistributionGroups": [],
        "channelType": "chat",
        "id": "5900",
        "maxTimeInQueue": 500,
        "name": "Chat_Automation_Queue",
        "routingType": null,
        "skillBasedRoutingType": null,
        "type": "inboundqueue"
      }, {
        "analyzerId": "AXDiu0LjKin420v3QJXN",
        "callDistributionGroups": [],
        "channelType": "chat",
        "id": "3381",
        "maxTimeInQueue": 500,
        "name": "Dont_delete_Chat_queue",
        "routingType": null,
        "skillBasedRoutingType": null,
        "type": "inboundqueue"
      }, {
        "analyzerId": "AXCZ1Cc0B023QCbPLXpr",
        "callDistributionGroups": [],
        "channelType": "chat",
        "id": "3270",
```

```

        "maxTimeInQueue": 500,
        "name": "Chat_Queue",
        "routingType": null,
        "skillBasedRoutingType": null,
        "type": "inboundqueue"
    }}
  },
  "jsMethod": "vteamListChanged"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_0e6eff89-165e-46b6-8c85-1aada01dab82",
"type": "VteamList"
}

```

## vTeamTransfer()

vTeam 転送リクエストを開始します。

### 例

```

await Desktop.agentContact.vteamTransfer({
  interactionId: "dab1a2f0 - bad0 - 11 ea - 8e7 c - 0 d99ede2535a",
  data: {
    vteamId: "AXDiuOLjKin420v3QJXN",
    vteamType: "inboundqueue"
  }
});

// Vteam Transfer Payload Type

{
  interactionId: string;
  data: Contact.vteamTransferPayLoad
}

type vteamTransferPayLoad = {
  vteamId: string;
  vteamType: string;
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	vTeam 転送リクエストのオプション。	はい
-->vteamId	文字列	vTeam の一意の識別子。	はい
-->vteamType	文字列	vTeam のタイプ。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。





- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentVteamTransferred`、そうでない場合は `AgentVteamTransferFailed` です。

### 応答の例

```
const vTeamTransferResponse = {
  "agentId": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
  "eventType": "RoutingMessage",
  "interaction": {
    "callAssociatedData": {
      "ani": {
        "value": "janedoe@gmail.com"
      },
      "category": {
        "value": "Sales"
      },
      "customerName": {
        "value": "Jane Doe"
      },
      "dn": {
        "value": "Desktop"
      },
      "entryPointId": {
        "value": "AXCqFyz1G2pKap9FI-mW"
      },
      "guestId": {
        "value":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTRlODEtOGMxMilmZjkyMTE3MmQ1NzY"
      },
      "mediaChannel": {
        "value": "web"
      },
      "reason": {
        "value": "Test"
      },
      "reasonCode": {
        "value": "Sales"
      },
      "ronaTimeout": {
        "value": "30"
      },
      "roomTitle": {
        "value": "Help Jane Doe with Sales"
      },
      "taskToBeSelfServiced": {
        "value": "false"
      },
      "templateId": {
        "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
      },
      "templateName": {
        "value": "Desktop"
      },
      "virtualTeamName": {
        "value": "Chat_Queue"
      }
    },
    "callAssociatedDetails": {
      "ani": "janedoe@gmail.com",
```

```

        "category": "Sales",
        "customerName": "Jane Doe",
        "dn": "Desktop",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTR1ODEtOGMxMilmZjkyMTE3MmQ1NzY",
        "mediaChannel": "web",
        "reason": "Test",
        "reasonCode": "Sales",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "virtualTeamName": "Chat_Queue"
    },
    "callFlowParams": {
        "Automation": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Automation",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Debit": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Debit",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Sales": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Sales",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        }
    },
    "callProcessingDetails": {
        "QMgrName": "aqm",
        "QueueId": "3270",
        "ani": "janedoe@gmail.com",
        "category": "Sales",
        "customerName": "Jane Doe",
        "dnis": "Desktop",
        "doNotRouteToAgents": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
"Y21zY29zcGFyazovL3VzL1BFT1BMRS81NzNiODg1MilkZjM3LTR1ODEtOGMxMilmZjkyMTE3MmQ1NzY",
        "mediaChannel": "web",
        "pauseDuration": "10",
        "pauseResumeEnabled": "true",
        "reason": "Test",
        "reasonCode": "Sales",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Sales",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "tenantId": "133",
        "virtualTeamName": "Chat_Queue",
        "vteamId": "7178",
        "vteamType": "inboundqueue"
    }
}

```

```

    },
    "contactDirection": {
      "type": "INBOUND"
    },
    "currentVTeam": "7178",
    "interactionId": "29bafa9c-40b5-11eb-8606-edd437f1b927",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
      "Y21zY29zcGFyazovL3VzL1JPT00vMmNiMTMwMDAtNDBiNS0xMWViLWE4M2ItZWl4ZTQwMDk1MDk5": {
        "holdTimestamp": null,
        "isHold": false,
        "mType": "mainCall",
        "mediaMgr": "cmm",
        "mediaResourceId":
      "Y21zY29zcGFyazovL3VzL1JPT00vMmNiMTMwMDAtNDBiNS0xMWViLWE4M2ItZWl4ZTQwMDk1MDk5",
        "mediaType": "chat",
        "participants": ["janedoe@gmail.com",
      "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed"]
      }
    },
    "mediaChannel": "web",
    "mediaType": "chat",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": null,
    "participants": {
      "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed": {
        "channelId": "ccbffa6f-4322-4e06-850f-e0ab2915f238",
        "consultState": null,
        "consultTimestamp": null,
        "dn": "9997770110",
        "hasJoined": true,
        "id": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
        "isConsulted": false,
        "isWrapUp": true,
        "joinTimestamp": 1608243279477,
        "lastUpdated": 1608254153975,
        "name": "FirstName LastName",
        "pType": "Agent",
        "queueId": "3270",
        "queueMgrId": "aqm",
        "sessionId": "b8c096ab-f9a0-49be-9efd-f8c78e1a7061",
        "siteId": "473",
        "teamId": "962",
        "teamName": "Team X",
        "type": "Agent",
        "wrapUpTimestamp": 1608254153975
      },
      "janedoe@gmail.com": {
        "id": "janedoe@gmail.com",
        "pType": "Customer",
        "type": "Customer"
      }
    }
  },
  "previousVTeams": ["3270"],
  "state": "new"
},
"interactionId": "29bafa9c-40b5-11eb-8606-edd437f1b927",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "91480930-40ce-11eb-840f-7f9e29c6b481",
"type": "AgentVteamTransferred"

```

```

    },
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25", "trackingId":
"notifs_08102baa-3437-4c6d-812c-db4ce8755ea0", "type": "RoutingMessage"
}

```

## blindTransfer()

ブラインド転送リクエストを開始します。

### 例

```

await Desktop.agentContact.blindTransfer({
  interactionId: "08259 a2b - bacf - 11 ea - bdd9 - abbd16eaf1d5",
  data: {
    agentId: "f795f41f - 3782 - 44 fa - 97 a4 - c8b4dc029477",
    destAgentId: "299 b728a - d6f8 - 4934 - 8 fad - 577525 c0b7fc",
    mediaType: "chat",
    destAgentTeamId: "964",
    destAgentDN: "9997770095",
    destSiteId: "472"
  }
});

// BlindTransfer Payload Type

{
  interactionId: string; data: Contact.blindTransferPayLoad
}

type blindTransferPayLoad = {
  agentId: string;
  destAgentId: string;
  mediaType: string;
  destAgentTeamId: string;
  destAgentDN: string;
  destSiteId: string;
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意的識別子。	はい
data	オブジェクト	ブラインド転送リクエストのオプション。	はい
-->agentId	文字列	エージェントの一意的識別子。	はい
-->destAgentId	文字列	ブラインド転送リクエストの接続先であるエージェントの一意的識別子。	はい
-->mediaType	文字列	チャットなどのメディアチャネルタイプ。	はい
-->destAgentTeamId	文字列	接続先エージェントが属するチームの一意的識別子。	はい
-->destAgentDN	文字列	接続先エージェントのダイヤル番号。	はい

名前	タイプ	説明	必須
-->destSiteId	文字列	接続先サイトの一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentBlindTransferred、そうでない場合は AgentBlindTransferFailedEvent です。

### 応答の例

```
const blindTransferResponse = {
  "data": {
    "agentId": "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2",
    "destAgentId": "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "agentEditable": false,
          "displayName": "ani",
          "name": "ani",
          "type": "STRING",
          "value": "displayName"
        },
        "appUser": {
          "agentEditable": false,
          "displayName": "appUser",
          "name": "appUser",
          "type": "STRING",
          "value": "qaus1-appuser-1212"
        },
        "customerName": {
          "agentEditable": false,
          "displayName": "customerName",
          "name": "customerName",
          "type": "STRING",
          "value": "displayName"
        },
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "AXVP21xF0G6K4V3pFKRG"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        }
      }
    }
  }
}
```

```

        "virtualTeamName": {
            "agentEditable": false,
            "displayName": "virtualTeamName",
            "name": "virtualTeamName",
            "type": "STRING",
            "value": "QA-e2e-SocialQueue"
        }
    },
    "callAssociatedDetails": {
        "ani": "*****",
        "appUser": "appuser-1212",
        "customerName": "displayName",
        "dn": "*****",
        "ronaTimeout": "30",
        "virtualTeamName": "QA-e2e-SocialQueue"
    },
    "callFlowParams": {},
    "callProcessingDetails": {
        "BLIND_TRANSFER_IN_PROGRESS": "true",
        "QMgrName": "aqm",
        "QueueId": "5134",
        "ani": "*****",
        "appUser": "qaus1-appuser-1212",
        "customerName": "displayName",
        "dnis": "*****",
        "pauseDuration": "30",
        "pauseResumeEnabled": "true",
        "ronaTimeout": "30",
        "taskToBeSelfServiced": "false",
        "tenantId": "82",
        "virtualTeamName": "SocialQueue",
        "vteamId": "5134"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "5134",
    "interactionId": "1da228dc-66cc-11eb-b852-ed71847d1aec",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
        "Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1": {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "mainCall",
            "mediaMgr": "socialmm",
            "mediaResourceId":
                "Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMWVhLTk0YjAtNzdjZDBkYWVjNDA1",
            "mediaType": "social",
            "participants": [
                "displayName"
            ]
        }
    },
    "mediaChannel": "type",
    "mediaType": "social",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf",
    "participants": {
        "337c1cca-d79b-44fc-8d8e-20d1b5e4dadf": {
            "channelId": "e08a6068-688d-472b-887b-1c5d39c7155f",
            "consultState": null,

```

```

        "consultTimestamp": null,
        "dn": "*****",
        "hasJoined": false,
        "id": "337c1cca-d79b-44fc-8d8e-20dlb5e4dadf",
        "isConsulted": false,
        "isWrapUp": false,
        "joinTimestamp": null,
        "lastUpdated": 1612431250963,
        "name": "social-agent2",
        "pType": "Agent",
        "queueId": "5134",
        "queueMgrId": "aqm",
        "sessionId": "a9aa9867-e62f-4415-9ea4-6babace39d4f",
        "siteId": "205",
        "teamId": "384",
        "teamName": "TeamSales",
        "type": "Agent",
        "wrapUpTimestamp": null
    },
    "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2": {
        "channelId": "59826e62-08c8-41e6-b51c-4a2b6d81d2f6",
        "consultState": null,
        "consultTimestamp": null,
        "dn": "*****",
        "hasJoined": true,
        "id": "cdf786ef-3285-4b2f-8ef9-3cf8a4c755c2",
        "isConsulted": false,
        "isWrapUp": true,
        "joinTimestamp": 1612431249042,
        "lastUpdated": 1612431251178,
        "name": "qaus1-gt-social-agent1 qaus1-gt-social-agent1",
        "pType": "Agent",
        "queueId": "5134",
        "queueMgrId": "aqm",
        "sessionId": "600e7ebb-20aa-4c91-a4f0-ebc596249603",
        "siteId": "205",
        "teamId": "384",
        "teamName": "TeamSales",
        "type": "Agent",
        "wrapUpTimestamp": 1612431251178
    },
    "displayName": {
        "id": "displayName",
        "pType": "Customer",
        "type": "Customer"
    }
},
"previousVTeams": [
    "AXVP21xF0G6K4V3pFKRG"
],
"state": "new",
"workflowManager": null
},
"interactionId": "1da228dc-66cc-11eb-b852-ed71847d1aec",
"mediaResourceId":
"Y21zY29zcGFyazovL3VzL1JPT00vNGQ5NGJmMjAtYjYwZi0xMwVhLTk0YjAtNzdjZDBkYWVjNDAl",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "4a958ace-49a2-49aa-b2e5-8798ede03f13",
"type": "AgentBlindTransferred"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_68c519c8-04f5-48b4-a5cf-609a09ccd765",

```

```
    "type": "RoutingMessage"
  }
}
```

## consultTransfer()

打診転送リクエストを開始します。

### 例

```
await Desktop.agentContact.consultTransfer({
  interactionId: "5 c3d487a - 874 b - 447 d - b5f2 - celf626301f5",
  data: {
    agentId: "43 bfcdef - 5551 - 45 b1 - a8f9 - a0b33e66e3fe",
    destAgentId: "f795f41f - 3782 - 44 fa - 97 a4 - c8b4dc029477",
    mediaType: "telephony",
    mediaResourceId: "b102ed10 - fac2 - 4 f8e - bece - 1 c2da6ba6dd8",
    destinationType: "Agent",
  }
});

// consultTransfer Payload Type

{
  interactionId: string; data: Contact.consultTransferPayload
}

type consultTransferPayload = {
  agentId ? : string | undefined;
  destAgentId: string;
  mediaType: string;
  mediaResourceId: string;
  destinationType ? : string | undefined;
}
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	打診転送リクエストのオプション。	はい
-->agentId	文字列	エージェントの一意の識別子。	はい
-->destAgentId	文字列	打診転送リクエストの接続先であるエージェントの一意の識別子。	はい
-->mediaType	文字列	テレフォニーなどのメディアチャンネルタイプ。	はい
-->mediaResourceId	文字列	メディアリソースの一意の識別子。	はい
-->destinationType	文字列	送信する打診転送リクエストのタイプ。接続先タイプには、エージェント、キュー、またはDNを使用できます。	はい



## 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は AgentConsultTransferred、そうでない場合は AgentConsultTransferFailed です。

## 応答の例

```
const consultTransferResponse = {
  "data": {
    "agentId": "ff7dbe19-fa3f-4c63-8a1c-04dcffef8e4f",
    "consultMediaResourceId": "fd0af60c-3250-46c3-96f8-1921f6ecf170",
    "destAgentId": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
    "destinationType": "Agent",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "": {
          "agentEditable": false,
          "displayName": "",
          "name": "",
          "type": "STRING",
          "value": ""
        },
        "IvrPath": {
          "agentEditable": false,
          "displayName": "IvrPath",
          "name": "IvrPath",
          "type": "STRING",
          "value": " EOI"
        },
        "ani": {
          "agentEditable": false,
          "displayName": "ani",
          "name": "ani",
          "type": "STRING",
          "value": "*****"
        },
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "*****"
        },
        "nextVteamId": {
          "agentEditable": false,
          "displayName": "nextVteamId",
          "name": "nextVteamId",
          "type": "STRING",
          "value": "Queue"
        },
        "pathId": {
          "agentEditable": false,
          "displayName": "pathId",
          "name": "pathId",
          "type": "STRING",
          "value": " StartCall PlayDone"
        }
      }
    }
  }
}
```

```

    },
    "ronaTimeout": {
      "agentEditable": false,
      "displayName": "ronaTimeout",
      "name": "ronaTimeout",
      "type": "STRING",
      "value": "30"
    },
    "virtualTeamName": {
      "agentEditable": false,
      "displayName": "virtualTeamName",
      "name": "virtualTeamName",
      "type": "STRING",
      "value": "Queue-1"
    }
  },
  "callAssociatedDetails": {
    "": "",
    "ivrPath": "EOI",
    "ani": "*****",
    "dn": "*****",
    "nextVteamId": "Queue",
    "pathId": " StartCall PlayDone",
    "ronaTimeout": "30",
    "virtualTeamName": "Queue-1"
  },
  "callFlowParams": {
    "Queue": {
      "description": "(vteam, A valid VTeam.)",
      "name": "Queue",
      "qualifier": "vteam",
      "value": "1336",
      "valueDataType": "string"
    },
    "WelcomePrompt": {
      "description": "(mediaFile, A valid media file.)",
      "name": "WelcomePrompt",
      "qualifier": "mediaFile",
      "value": "GAwelcome.wav",
      "valueDataType": "string"
    }
  },
  "callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "1336",
    "ani": "*****",
    "dnis": "*****",
    "isPaused": "false",
    "jscriptId": "AW7HP3QNB44q0sL-cTVz",
    "pauseDuration": "30",
    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "82",
    "virtualTeamName": "Queue-1",
    "vteamId": "1336"
  },
  "contactDirection": {
    "type": "INBOUND"
  },
  "currentVTeam": "1336",
  "interactionId": "1277da79-dc49-439c-ab72-f00453a1ff71",
  "isFcManaged": false,

```

```

"isTerminated": false,
"media": {
  "1277da79-dc49-439c-ab72-f00453a1ff71": {
    "holdTimestamp": null,
    "isHold": false,
    "mType": "mainCall",
    "mediaMgr": "vmm",
    "mediaResourceId": "1277da79-dc49-439c-ab72-f00453a1ff71",
    "mediaType": "telephony",
    "participants": [
      "*****",
      "9b036d89-930c-4187-a5bd-5dbb1439fe41"
    ]
  }
},
"mediaChannel": "broadcloud",
"mediaType": "telephony",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"outboundType": null,
"owner": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
"participants": {
  "*****": {
    "id": "*****",
    "pType": "Customer",
    "type": "Customer"
  },
  "9b036d89-930c-4187-a5bd-5dbb1439fe41": {
    "channelId": "3de6706f-95b7-485a-9304-30928f7ee52e",
    "consultState": null,
    "consultTimestamp": null,
    "dn": "*****",
    "hasJoined": true,
    "id": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
    "isConsulted": true,
    "isWrapUp": false,
    "joinTimestamp": 1612429549032,
    "lastUpdated": 1612429549032,
    "name": "qaus1-gt-user1 qaus1-gt-user1",
    "pType": "Agent",
    "queueId": "1336",
    "queueMgrId": "aqm",
    "sessionId": "90896c91-0f77-4bc6-aec3-488a34ca88d3",
    "siteId": "205",
    "teamId": "598",
    "teamName": "Sales",
    "type": "Agent",
    "wrapUpTimestamp": null
  },
  "ff7dbel9-fa3f-4c63-8a1c-04dcffef8e4f": {
    "channelId": "fb704337-71c3-4802-a03b-84a2d18eeaa0",
    "consultState": "consulting",
    "consultTimestamp": 1612429549052,
    "dn": "*****",
    "hasJoined": true,
    "id": "ff7dbel9-fa3f-4c63-8a1c-04dcffef8e4f",
    "isConsulted": false,
    "isWrapUp": false,
    "joinTimestamp": 1612429537991,
    "lastUpdated": 1612429537991,
    "name": "qaus1-gt-user2 q",
    "pType": "Agent",
    "queueId": "1336",
    "queueMgrId": "aqm",
    "sessionId": "57323f8b-9460-45fa-aaf6-4ca51c327af8",

```

```

        "siteId": "205",
        "teamId": "598",
        "teamName": "Sales",
        "type": "Agent",
        "wrapUpTimestamp": null
      }
    },
    "previousVTeams": [
      "AW6hlAVRrZSGni185x28"
    ],
    "state": "connected",
    "workflowManager": null
  },
  "interactionId": "1277da79-dc49-439c-ab72-f00453a1ff71",
  "mediaResourceId": "1277da79-dc49-439c-ab72-f00453a1ff71",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "queueMgr": "aqm",
  "trackingId": "05b38615-5bdc-4e1d-8efe-60eal70fd5c",
  "transferredMediaResourceId": "1277da79-dc49-439c-ab72-f00453a1ff71",
  "type": "AgentConsultTransferred"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_3464be63-83e7-4b2d-94d5-a63fddc108d5",
"type": "RoutingMessage"
}
}

```

## hold()

音声コールを保留します。

### 例

```

await Desktop.agentContact.hold({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3",
  data: {
    mediaResourceId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
  }
});

// Hold Payload

{
  interactionId: string;data: {
    mediaResourceId: string
  }
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	音声コールを保留するオプション。	はい
->mediaResourceId	文字列	メディアリソースの一意の識別子。	はい

## 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentContactHeld`、そうでない場合は `AgentContactHoldFailed` です。

## 応答の例

```
const holdResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "destAgentId": null,
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "8895579172"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        },
        "virtualTeamName": {
          "agentEditable": false,
          "displayName": "virtualTeamName",
          "name": "virtualTeamName",
          "type": "STRING",
          "value": "Outdial Queue-1"
        }
      },
      "callAssociatedDetails": {
        "dn": "8895579172",
        "ronaTimeout": "30",
        "virtualTeamName": "Outdial Queue-1"
      },
      "callFlowParams": {
        "OutdialQueue": {
          "description": "(vteam, The Outdial Queue.)",
          "name": "OutdialQueue",
          "qualifier": "vteam",
          "value": "3264",
          "valueDataType": "string"
        }
      },
      "callProcessingDetails": {
        "QMgrName": "aqm",
        "QueueId": "3264",
        "dnis": "8895579172",
        "isPaused": "false",
        "outdialTransferToQueueEnabled": "false",
        "pauseDuration": "10",
```

```

    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "virtualTeamName": "Outdial Queue-1",
    "vteamId": "AXCLfZZU9S1oTdqE1OFZ"
  },
  "contactDirection": {
    "type": "OUTBOUND"
  },
  "currentVTeam": "3264",
  "interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
  "isFcManaged": false,
  "isTerminated": false,
  "media": {
    "1dd08726-bca5-4ce8-9b51-b20dca8ab154": {
      "holdTimestamp": 1612355834441,
      "isHold": true,
      "mType": "mainCall",
      "mediaMgr": "vmm",
      "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
      "mediaType": "telephony",
      "participants": [
        ""
      ]
    }
  },
  "mediaChannel": "dialer",
  "mediaType": "telephony",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": "OUTDIAL",
  "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
  "participants": {
    "": {
      "id": "",
      "pType": "Customer",
      "type": "Customer"
    },
    "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
      "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "8895579172",
      "hasJoined": true,
      "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1612354635705,
      "lastUpdated": 1612354635705,
      "name": "John Doe",
      "pType": "Agent",
      "queueId": "3264",
      "queueMgrId": "aqm",
      "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
      "siteId": "472",
      "teamId": "960",
      "teamName": "Email_Team",
      "type": "Agent",
      "wrapUpTimestamp": null
    }
  },
  "previousVTeams": [

```

```

    ],
    "state": "connected",
    "workflowManager": null
  },
  "interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
  "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "queueMgr": "aqm",
  "trackingId": "3768264d-8588-465f-9ee6-0c25418ea9f7",
  "type": "AgentContactHeld"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_8e8cabb7-6d5a-460d-9b8b-ecfad96448b3",
"type": "RoutingMessage"
}

```

## unHold()

保留された音声コールを再開します。

### 例

```

await Desktop.agentContact.unHold({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3",
  data: {
    mediaResourceId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
  }
});

// unHold Payload Type

{
  interactionId: string;data: {
    mediaResourceId: string
  }
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	保留されている音声コールを再開するオプション。	はい
→mediaResourceId	文字列	メディアリソースの一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `AgentContactUnheld`、そうでない場合は `AgentContactUnholdFailed` です。

## 応答の例

```

const unHoldResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "destAgentId": null,
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "8895579172"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        },
        "virtualTeamName": {
          "agentEditable": false,
          "displayName": "virtualTeamName",
          "name": "virtualTeamName",
          "type": "STRING",
          "value": "Outdial Queue-1"
        }
      }
    },
    "callAssociatedDetails": {
      "dn": "8895579172",
      "ronaTimeout": "30",
      "virtualTeamName": "Outdial Queue-1"
    },
    "callFlowParams": {
      "OutdialQueue": {
        "description": "(vteam, The Outdial Queue.)",
        "name": "OutdialQueue",
        "qualifier": "vteam",
        "value": "3264",
        "valueDataType": "string"
      }
    },
    "callProcessingDetails": {
      "QMgrName": "aqm",
      "QueueId": "3264",
      "dnis": "8895579172",
      "isPaused": "false",
      "outdialTransferToQueueEnabled": "false",
      "pauseDuration": "10",
      "pauseResumeEnabled": "true",
      "recordInProgress": "true",
      "ronaTimeout": "30",
      "taskToBeSelfServiced": "false",
      "tenantId": "133",
      "virtualTeamName": "Outdial Queue-1",
      "vteamId": "AXCLfZZU9S1oTdqe10FZ"
    },
    "contactDirection": {
      "type": "OUTBOUND"
    },
    "currentVTeam": "3264",
  }
}

```



```

"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"isFcManaged": false,
"isTerminated": false,
"media": {
  "1dd08726-bca5-4ce8-9b51-b20dca8ab154": {
    "holdTimestamp": null,
    "isHold": false,
    "mType": "mainCall",
    "mediaMgr": "vmm",
    "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
    "mediaType": "telephony",
    "participants": [
      ""
    ]
  }
},
"mediaChannel": "dialer",
"mediaType": "telephony",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"outboundType": "OUTDIAL",
"owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
"participants": {
  "": {
    "id": "",
    "pType": "Customer",
    "type": "Customer"
  },
  "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
    "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
    "consultState": null,
    "consultTimestamp": null,
    "dn": "8895579172",
    "hasJoined": true,
    "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "isConsulted": false,
    "isWrapUp": false,
    "joinTimestamp": 1612354635705,
    "lastUpdated": 1612354635705,
    "name": "John Doe",
    "pType": "Agent",
    "queueId": "3264",
    "queueMgrId": "aqm",
    "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
    "siteId": "472",
    "teamId": "960",
    "teamName": "Email_Team",
    "type": "Agent",
    "wrapUpTimestamp": null
  }
},
"previousVTeams": [
],
"state": "connected",
"workflowManager": null
},
"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "a5152cc8-004a-4e4c-b5d8-c9fc947f1e57",
"type": "AgentContactUnheld"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",

```

## pauseRecording()

```

    "trackingId": "notifs_85f4d943-8e95-49c9-92c4-66047d526506",
    "type": "RoutingMessage"
  }

```

## pauseRecording()

顧客からクレジットカード情報などの機密性の高い情報を取得する間、コールの録音を一時停止します。

管理者が構成している場合、さまざまな理由で音声コールが録音されます。エージェントが機密性の高い情報を処理するときに、録音の一時停止をエージェントが判断する場合があります。一時停止/再開機能は、Agent Desktopのインタラクション制御 ペインから使用できます。ウィジェットから一時停止/再開機能にアクセスすることもできます。

### 例

```

await Desktop.agentContact.pauseRecording({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3"
});
// Pausing a recording for telephony task

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の type パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は ContactRecordingPaused、そうでない場合は ContactRecordingPauseFailed です。

### 応答の例

```

const pauseRecordingResponse = {
  "data": {
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "8895579172"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",

```

```

        "value": "30"
    },
    "virtualTeamName": {
        "agentEditable": false,
        "displayName": "virtualTeamName",
        "name": "virtualTeamName",
        "type": "STRING",
        "value": "Outdial Queue-1"
    }
},
"callAssociatedDetails": {
    "dn": "8895579172",
    "ronaTimeout": "30",
    "virtualTeamName": "Outdial Queue-1"
},
"callFlowParams": {
    "OutdialQueue": {
        "description": "(vteam, The Outdial Queue.)",
        "name": "OutdialQueue",
        "qualifier": "vteam",
        "value": "3264",
        "valueDataType": "string"
    }
},
"callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3264",
    "dnis": "8895579172",
    "isPaused": "true",
    "outdialTransferToQueueEnabled": "false",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "virtualTeamName": "Outdial Queue-1",
    "vteamId": "AXCLfZZU9S1oTdqe1OFZ"
},
"contactDirection": {
    "type": "OUTBOUND"
},
"currentVTeam": "3264",
"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"isFcManaged": false,
"isTerminated": false,
"media": {
    "1dd08726-bca5-4ce8-9b51-b20dca8ab154": {
        "holdTimestamp": null,
        "isHold": false,
        "mType": "mainCall",
        "mediaMgr": "vmm",
        "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
        "mediaType": "telephony",
        "participants": [
            ""
        ]
    }
},
"mediaChannel": "dialer",
"mediaType": "telephony",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"outboundType": "OUTDIAL",
"owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",

```

```

    "participants": {
      "": {
        "id": "",
        "pType": "Customer",
        "type": "Customer"
      },
      "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
        "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
        "consultState": null,
        "consultTimestamp": null,
        "dn": "8895579172",
        "hasJoined": true,
        "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
        "isConsulted": false,
        "isWrapUp": false,
        "joinTimestamp": 1612354635705,
        "lastUpdated": 1612354635705,
        "name": "John Doe",
        "pType": "Agent",
        "queueId": "3264",
        "queueMgrId": "aqm",
        "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
        "siteId": "472",
        "teamId": "960",
        "teamName": "Email_Team",
        "type": "Agent",
        "wrapUpTimestamp": null
      }
    },
    "previousVTeams": [
    ],
    "state": "connected",
    "workflowManager": null
  },
  "interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "queueMgr": "aqm",
  "trackingId": "b031f6d7-63a3-49d9-b377-e848bce41bd3",
  "type": "ContactRecordingPaused"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_52a00992-95e8-438c-ac44-8302da18d252",
"type": "RoutingMessage"
}

```

## resumeRecording()

一時停止した音声コールの録音を再開します。

### 例

```

await Desktop.agentContact.resumeRecording({
  interactionId: "c837e6f7 - 699 a - 4736 - bb82 - 03 a6d58bb7f3",
  data: {
    autoResumed: true
  }
});
// Resuming recording for telephony task
// resumeRecording Payload Type

{
  interactionId: string;data: {

```

```

    autoResumed: boolean
  }
}

```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
interactionId	文字列	ユーザーインタラクションの一意の識別子。	はい
data	オブジェクト	一時停止した音声コールの録音を再開するオプション。	はい
->autoResumed	ブール値	一時停止したコールを手動で再開する必要があるかどうかを決定します。  <ul style="list-style-type: none"> <li>• <b>True</b> : 指定時間の経過後に自動的に音声コールが自動的に再開されます。</li> <li>• <b>False</b> : 一時停止された音声コールに対して、定義されたボタンに基づいてユーザーがアクションを実行する必要があります。たとえば、[録音開始 (Start Recording)] です。</li> </ul>	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。



- (注) レスポンスペイロード内の `type` パラメータ値は、このメソッドが成功か失敗かによって決まります。メソッドが成功した場合は、値は `ContactRecordingResumed`、そうでない場合は `ContactRecordingResumeFailed` です。

### 応答の例

```

const resumeRecordingResponse = {
  "data": {
    "autoResumed": true,
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "8895579172"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        }
      }
    }
  }
}

```

```

    },
    "virtualTeamName": {
      "agentEditable": false,
      "displayName": "virtualTeamName",
      "name": "virtualTeamName",
      "type": "STRING",
      "value": "Outdial Queue-1"
    }
  },
  "callAssociatedDetails": {
    "dn": "8895579172",
    "ronaTimeout": "30",
    "virtualTeamName": "Outdial Queue-1"
  },
  "callFlowParams": {
    "OutdialQueue": {
      "description": "(vteam, The Outdial Queue.)",
      "name": "OutdialQueue",
      "qualifier": "vteam",
      "value": "3264",
      "valueDataType": "string"
    }
  },
  "callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3264",
    "dnis": "8895579172",
    "isPaused": "false",
    "outdialTransferToQueueEnabled": "false",
    "pauseDuration": "10",
    "pauseResumeEnabled": "true",
    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "virtualTeamName": "Outdial Queue-1",
    "vteamId": "AXCLFZZU9S1oTdqE1OFZ"
  },
  "contactDirection": {
    "type": "OUTBOUND"
  },
  "currentVTeam": "3264",
  "interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
  "isFcManaged": false,
  "isTerminated": false,
  "media": {
    "1dd08726-bca5-4ce8-9b51-b20dca8ab154": {
      "holdTimestamp": null,
      "isHold": false,
      "mType": "mainCall",
      "mediaMgr": "vmm",
      "mediaResourceId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
      "mediaType": "telephony",
      "participants": [
        ""
      ]
    }
  },
  "mediaChannel": "dialer",
  "mediaType": "telephony",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": "OUTDIAL",
  "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
  "participants": {

```

```
    "": {
      "id": "",
      "pType": "Customer",
      "type": "Customer"
    },
    "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
      "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "8895579172",
      "hasJoined": true,
      "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1612354635705,
      "lastUpdated": 1612354635705,
      "name": "John Doe",
      "pType": "Agent",
      "queueId": "3264",
      "queueMgrId": "aqm",
      "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
      "siteId": "472",
      "teamId": "960",
      "teamName": "Email_Team",
      "type": "Agent",
      "wrapUpTimestamp": null
    }
  },
  "previousVTeams": [
  ],
  "state": "connected",
  "workflowManager": null
},
"interactionId": "1dd08726-bca5-4ce8-9b51-b20dca8ab154",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "9cf5bfc4-d120-4330-9996-039a9937c52c",
"type": "ContactRecordingResumed"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_c04510a0-140a-439c-9ab4-68b6fdf6ea2e",
"type": "RoutingMessage"
}
```

## 非同期イベント

非同期イベントをサブスクライブするには、次のイベントを使用します。

- eAgentContact
- eAgentContactAssigned
- eAgentContactAssignFailed
- eAgentContactWrappedUp
- eAgentOfferContact
- eAgentOfferContactRona
- eAgentOfferConsult
- eAgentWrapup
- eAgentContactHeld
- eAgentContactUnHeld
- eCallRecordingStarted
- eResumeRecording
- ePauseRecording
- eConsultTransfer
- eAgentblindTransferred
- eAgentvteamTransfer
- eAgentConsultCreated
- eAgentConsultConferenced
- eAgentConsultEnded
- eAgentCtqCancelled
- eAgentConsultConferenceEnded
- eAgentConsulting
- eAgentConsultFailed
- eAgentConsultEndFailed
- eAgentCtqFailed
- eAgentCtqCancelFailed
- eAgentConsultConferenceEndFailed

### 例

```
await Desktop.agentStateInfo.stateChange({
  state: "Available",
  auxCodeIdArray: "0",
});

// Get success/failure event here
Desktop.agentStateInfo.addEventListener("updated", updatedList =>
  logger.info(updatedList)
);
```

メモリリークを回避するためにイベントサブスクリプションを削除するには、以下のオプションを使用します。

```
// Module supports removing added listeners like:
const listener = msg => logger.info(msg);
Desktop.agentContact.addEventListener("eAgentContact", listener);
Desktop.agentContact.removeEventListener("eAgentContact", listener);

// Module supports one-time added listeners like:
Desktop.agentContact.addOnceEventListener("eAgentContact", listener);
Desktop.agentContact.removeOnceEventListener("eAgentContact", listener);

// Module supports removing all listeners like:
Desktop.agentContact.removeAllEventListeners();
```



(注) stateChange などの非同期イベントの応答よりも、イベントを使用することを推奨します。



## メソッド

### addEventListener(event, handler)

イベントリスナーを追加します。このメソッドは、特定のイベントが生成された場合に、そのイベントをリスンします。イベント名は、文字列としてパラメータに指定できます。

#### 例

```
Desktop.agentContact.addEventListener("eAgentContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

#### パラメータ

名前	タイプ	説明	必須
event	文字列	イベントのタイプ。	はい
handler	関数	条件と一致したときに呼び出される関数。	はい

### removeEventListener(event, handler)

[addEventListener\(event, handler\)](#) メソッドを使用して接続された、追加済みのイベントリスナーを削除します。

#### 例

```
Desktop.agentContact.removeEventListener("eAgentContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

#### パラメータ

名前	タイプ	説明	必須
event	文字列	イベントのタイプ。	はい
handler	関数	条件と一致したときに呼び出される関数。	はい

### addOnceEventListener(event, handler)

一時的なイベントリスナーを追加します。

#### 例

```
Desktop.agentContact.addOnceEventListener("eAgentContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

#### パラメータ

名前	タイプ	説明	必須
event	文字列	イベントのタイプ。	はい
handler	関数	条件と一致したときに呼び出される関数。	はい

## removeOnceEventListener(event, handler)

一時的に追加されたイベントリスナーを削除します。

### 例

```
Desktop.agentContact.removeOnceEventListener("eAgentContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

### パラメータ

名前	タイプ	説明	必須
event	文字列	イベントのタイプ。	はい
handler	関数	条件と一致したときに呼び出される関数。	はい

## removeAllEventListeners(event, handler)

追加されたイベントリスナーをすべて削除します。

### 例

```
Desktop.agentContact.removeAllEventListeners();
```

## イベント

### eAgentContact

エージェントタスクを含むデスクトップを読み込みます。

### 例

```
Desktop.agentContact.addEventListener("eAgentContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

### 応答の例

```
const eAgentContact = {
  type: 'RoutingMessage',
  orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
  trackingId: 'notifs_6f0ee13d-bb2f-4600-8631-c7d86fb08dfe',
  data: {
    mediaResourceId: '80c213d5-2747-4bf4-8a0c-9c45b0afdf2a',
    eventType: 'RoutingMessage',
    agentId: '9738292f-3a68-4909-bdea-b541cccd6935',
    trackingId: '68bca42c-8a50-4cf1-8c18-f5229b04afa9',
    interaction: {
      isFcManaged: false,
      isTerminated: false,
      mediaType: 'telephony',
      previousVTeams: ['AXCZ0YCAxrf9I0XFBI7b'],
      state: 'new',
      currentVTeam: '3268',
      participants: {
        '+19997770095': {
          id: '+19997770095',
          pType: 'Customer',
          type: 'Customer'
        }
      }
    }
  }
}
```

```

    },
    '9738292f-3a68-4909-bdea-b541cccd6935': {
      joinTimestamp: null,
      name: '',
      pType: 'Agent',
      teamName: 'Email_Team',
      lastUpdated: 1593440068730,
      teamId: '960',
      isConsulted: false,
      hasJoined: false,
      dn: '9997770093',
      queueId: '3268',
      id: '9738292f-3a68-4909-bdea-b541cccd6935',
      sessionId: '054ebc22-b34e-4e1b-8730-da294093f813',
      queueMgrId: 'aqm',
      siteId: '472',
      type: 'Agent',
      channelId: '53e05ff1-69c7-425f-a660-5f9e21f0a4bb',
      isWrapUp: false
    }
  },
  interactionId: '80c213d5-2747-4bf4-8a0c-9c45b0afdf2a',
  orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
  callProcessingDetails: {
    QMgrName: 'aqm',
    taskToBeSelfServiced: 'false',
    ani: '+19997770095',
    dnis: '+12147659000',
    tenantId: '133',
    QueueId: '3268',
    vteamId: '3268',
    jsriptId: 'AXCZ4c3mjkkgAuS7vSIU',
    virtualTeamName: 'Queue - Telephony',
    customerName: 'Jane Doe',
    ronaTimeout: '30',
    reasonCode: 'Credit',
    IvrPath: 'EOI',
    pathId: ' StartCall PlayDone out out out'
  },
  media: {
    '80c213d5-2747-4bf4-8a0c-9c45b0afdf2a': {
      mediaResourceId: '80c213d5-2747-4bf4-8a0c-9c45b0afdf2a',
      mediaType: 'telephony',
      mediaMgr: 'vmm',
      participants: ['+19997770095'],
      mType: 'mainCall',
      isHold: false
    }
  },
  owner: '9738292f-3a68-4909-bdea-b541cccd6935',
  mediaChannel: 'broadcloud',
  contactDirection: {
    type: 'INBOUND'
  },
  callFlowParams: {
    Play2: {
      name: 'Play2',
      qualifier: '',
      description: '(A valid text.)',
      valueDataType: 'string',
      value: 'Welcome to Contact Center'
    },
    Queue3: {
      name: 'Queue3',

```

```

        qualifier: 'vteam',
        description: '(vteam, A valid VTeam.)',
        valueDataType: 'string',
        value: '3268'
      }
    },
    interactionId: '80c213d5-2747-4bf4-8a0c-9c45b0afdf2a',
    orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
    queueMgr: 'aqm',
    type: 'AgentContact',
    isConsulted: false,
    consultMediaResourceId: null
  }
};

```

## eAgentContactAssigned

新しいタスクまたはコンタクトが正常に割り当てられました。

### 例

```
Desktop.agentContact.addEventListener("eAgentContactAssigned", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [accept\(\)](#)

## eAgentContactAssignFailed

割り当てられたタスクまたはコンタクトは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentContactAssignFailed", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [accept\(\)](#)

## eAgentContactWrappedUp

受け付けたタスクまたはコンタクトはラップアップ済みです。

### 例

```
Desktop.agentContact.addEventListener("eAgentContactWrappedUp", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [wrapup\(\)](#)

## eAgentOfferContact

新しいタスクまたはコンタクトがエージェントにルーティングされました。

### 例

```
Desktop.agentContact.addEventListener("eAgentOfferContact", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

### 応答の例

```
const eAgentOfferContact = {
  "data": {
```

```

"agentId": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
"eventType": "RoutingMessage",
"interaction": {
  "callAssociatedData": {
    "ani": {
      "value": "janedoe@gmail.com"
    },
    "category": {
      "value": "Automation"
    },
    "customerName": {
      "value": "Jane Doe"
    },
    "dn": {
      "value": "Desktop"
    },
    "entryPointId": {
      "value": "AXCqFyz1G2pKap9PI-mW"
    },
    "guestId": {
      "value":
"Y21zY29zcGFyazovL3VzL1BFTlBMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MTVmZGIwMTA"
    },
    "mediaChannel": {
      "value": "web"
    },
    "reason": {
      "value": "Test"
    },
    "reasonCode": {
      "value": "Automation"
    },
    "ronaTimeout": {
      "value": "30"
    },
    "roomTitle": {
      "value": "Help Jane Doe with Automation"
    },
    "taskToBeSelfServiced": {
      "value": "false"
    },
    "templateId": {
      "value": "b57990c0-5ec6-11ea-96ee-5df5aef56329"
    },
    "templateName": {
      "value": "Desktop"
    },
    "virtualTeamName": {
      "value": "Chat_Queue"
    }
  },
  "callAssociatedDetails": {
    "ani": "janedoe@gmail.com",
    "category": "Automation",
    "customerName": "Jane Doe",
    "dn": "Desktop",
    "entryPointId": "AXCqFyz1G2pKap9PI-mW",
    "guestId":
"Y21zY29zcGFyazovL3VzL1BFTlBMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MTVmZGIwMTA",
    "mediaChannel": "web",
    "reason": "Test",
    "reasonCode": "Automation",
    "ronaTimeout": "30",
    "roomTitle": "Help Jane Doe with Automation",
  }
}

```

```

        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "virtualTeamName": "Chat_Queue"
    },
    "callFlowParams": {
        "Automation": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Automation",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Debit": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Debit",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        },
        "Sales": {
            "description": "(vteam, A valid VTeam.)",
            "name": "Sales",
            "qualifier": "vteam",
            "value": "3270",
            "valueDataType": "string"
        }
    },
    "callProcessingDetails": {
        "QMgrName": "aqm",
        "QueueId": "3270",
        "ani": "janedoe@gmail.com",
        "category": "Automation",
        "customerName": "Jane Doe",
        "dnis": "Desktop",
        "entryPointId": "AXCqFyz1G2pKap9PI-mW",
        "guestId":
        "Y21zY29zcGFyazovL3VzL1BFT1BMRS83ZjA3YjJmMC1jNTMyLTQ5MDktYTtkzNi0yYTA3MmVmZGIwMTA",
        "mediaChannel": "web",
        "reason": "Test",
        "reasonCode": "Automation",
        "ronaTimeout": "30",
        "roomTitle": "Help Jane Doe with Automation",
        "taskToBeSelfServiced": "false",
        "templateId": "b57990c0-5ec6-11ea-96ee-5df5aef56329",
        "templateName": "Desktop",
        "tenantId": "133",
        "virtualTeamName": "Chat_Queue",
        "vteamId": "AXCqFyz1G2pKap9PI-mW"
    },
    "contactDirection": {
        "type": "INBOUND"
    },
    "currentVTeam": "3270",
    "interactionId": "8d3a56fa-4172-11eb-abaf-1b20df734401",
    "isFcManaged": false,
    "isTerminated": false,
    "media": {
        "Y21zY29zcGFyazovL3VzL1JPT00vOTAxMmEzZjAtNDE3Mi0xMWViLWE1N2QtZWY2N2MwZTMmYmFh": {
            "holdTimestamp": null,
            "isHold": false,
            "mType": "mainCall",
            "mediaMgr": "cmm",

```

```

        "mediaResourceId":
        "Y21zY29zcGFyazovL3VzL1JPT00vOTAxMmEzZjAtNDE3Mi0xMWVlLWE1N2QtZWY2N2MwZTMwYmFh",
        "mediaType": "chat",
        "participants": ["janedoe@gmail.com"]
    },
    },
    "mediaChannel": "web",
    "mediaType": "chat",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": null,
    "owner": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
    "participants": {
        "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed": {
            "channelId": "bff9350a-9d2a-489a-a8d5-d9dacea4b692",
            "consultState": null,
            "consultTimestamp": null,
            "dn": "9997770110",
            "hasJoined": false,
            "id": "7d12d9ea-e8e0-41ee-81bf-c11a685b64ed",
            "isConsulted": false,
            "isWrapUp": false,
            "joinTimestamp": null,
            "lastUpdated": 1608324587206,
            "name": "uui-agent4 uui-agent4",
            "pType": "Agent",
            "queueId": "3270",
            "queueMgrId": "aqm",
            "sessionId": "102d2096-bcc0-45bb-a583-a02f6c188071",
            "siteId": "473",
            "teamId": "1940",
            "teamName": "Medical Help",
            "type": "Agent",
            "wrapUpTimestamp": null
        },
        "janedoe@gmail.com": {
            "id": "janedoe@gmail.com",
            "pType": "Customer",
            "type": "Customer"
        }
    },
    },
    "previousVTeams": [],
    "state": "new"
},
"interactionId": "8d3a56fa-4172-11eb-abaf-1b20df734401",
"mediaResourceId":
"Y21zY29zcGFyazovL3VzL1JPT00vOTAxMmEzZjAtNDE3Mi0xMWVlLWE1N2QtZWY2N2MwZTMwYmFh",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"ronaTimeout": 30,
"trackingId": "429bb73d-7463-4661-94aa-a3230d60e88a",
"type": "AgentOfferContact"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_f23e73e4-b130-4bce-b110-1c1da7225ffa",
"type": "RoutingMessage"
}

```

## eAgentOfferContactRona

エージェントは、対応可能な最大時間内にタスクまたはコンタクトリクエストを受け付けできませんでした。エージェントの状態がシステムによって [対応可能 (Available)] から [RONA] に変更されます。

**例**

```
Desktop.agentContact.addEventListener("eAgentOfferContactRona", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [decline\(\)](#)

**eAgentOfferConsult**

受け付けたタスクまたはコンタクト打診リクエストがエージェントにルーティングされました。

**例**

```
Desktop.agentContact.addEventListener("eAgentOfferConsult", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

**応答の例**

```
export const eAgentOfferConsult = {
  type: 'RoutingMessage',
  orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
  trackingId: 'notifs_50e9d868-71c0-48f2-9880-4a12fc505988',
  data: {
    consultMediaResourceId:
'consult_Y2lzY29zcGFyazovL3VzL1JPT00vODZkMGQ2MDAtOTQ0Yy0xMwVhLWE4MjMtNzFjMjA3YWRLYjAx',
    eventType: 'RoutingMessage',
    agentId: '8546f083-74ae-4510-b2a6-c8e21f01689b',
    consultingAgentId: '9738292f-3a68-4909-bdea-b541cccd6935',
    trackingId: 'ca074768-9d2b-427e-86af-3745c2a8cc60',
    interaction: {
      isFcManaged: false,
      isTerminated: false,
      mediaType: 'chat',
      previousVTeams: [],
      state: 'consult',
      currentVTeam: '3270',
      participants: {
        'mohaksin@cisco.com': {
          id: 'mohaksin@cisco.com',
          pType: 'Customer',
          type: 'Customer'
        },
        '8546f083-74ae-4510-b2a6-c8e21f01689b': {
          name: '',
          pType: 'Agent',
          teamName: 'Dont_Use_Team',
          joinTimestamp: null,
          lastUpdated: 1589287209385,
          teamId: '964',
          isConsulted: true,
          hasJoined: false,
          dn: '9997770111',
          queueId: '3270',
          id: '8546f083-74ae-4510-b2a6-c8e21f01689b',
          sessionId: 'dead6d11-3d8f-42ea-87d9-b94d5f5b6370',
          queueMgrId: 'aqm',
          siteId: '472',
          type: 'Agent',
          channelId: 'a4297667-6d16-4d9d-ad2f-b21fee6b97fe',
          isWrapUp: false
        },
        '9738292f-3a68-4909-bdea-b541cccd6935': {
```



```

        name: 'vivek',
        pType: 'Agent',
        teamName: 'Dont_Use_Team',
        joinTimestamp: 1594295035415,
        lastUpdated: 1589287209385,
        teamId: '964',
        isConsulted: false,
        hasJoined: true,
        dn: '9997770111',
        queueId: '3270',
        id: '9738292f-3a68-4909-bdea-b541cccd6935',
        sessionId: 'dead6d11-3d8f-42ea-87d9-b94d5f5b6370',
        queueMgrId: 'aqm',
        siteId: '472',
        type: 'Agent',
        channelId: 'a4297667-6d16-4d9d-ad2f-b21fee6b97fe',
        isWrapUp: false
    }
},
interactionId: 'cd4f3721-944b-11ea-a5e5-bdac2bb8b026',
orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
callProcessingDetails: {
    QMgrName: 'aqm',
    taskToBeSelfServiced: 'false',
    ani: 'mohaksin@cisco.com',
    dnis: 'AgentX',
    tenantId: '133',
    QueueId: '3270',
    vteamId: 'AXCqFyzlG2pKap9PI-mW',
    virtualTeamName: 'Chat_Queue',
    customerName: 'Mohak',
    ronaTimeout: '30',
    reasonCode: 'Sales',
    reason: 'Lockdown challeeya',
    category: 'Sales'
},
media: {
    Y21zY29zcGFyazovL3VzL1JPT00vODZkMGQ2MDAtOTQ0Yy0xMwVhLWE4MjMtNzFjMjA3YWRLYjAx: {
        mediaResourceId:
        'Y21zY29zcGFyazovL3VzL1JPT00vODZkMGQ2MDAtOTQ0Yy0xMwVhLWE4MjMtNzFjMjA3YWRLYjAx',
        mediaType: 'chat',
        mediaMgr: 'cmm',
        participants: ['mohaksin@cisco.com'],
        mType: 'mainCall',
        isHold: false
    }
},
owner: '8546f083-74ae-4510-b2a6-c8e21f01689b',
mediaChannel: 'web',
contactDirection: {
    type: 'INBOUND'
},
callFlowParams: {
    Sales: {
        name: 'Sales',
        qualifier: 'vteam',
        description: '(vteam, A valid VTeam.)',
        valueType: 'string',
        value: '3270'
    },
    Debit: {
        name: 'Debit',
        qualifier: 'vteam',

```

```

        description: '(vteam, A valid VTeam.)',
        valueDataType: 'string',
        value: '3270'
    },
    Automation: {
        name: 'Automation',
        qualifier: 'vteam',
        description: '(vteam, A valid VTeam.)',
        valueDataType: 'string',
        value: '3270'
    }
},
},
ronaTimeout: 30,
interactionId: 'cd4f3721-944b-11ea-a5e5-bdac2bb8b026',
orgId: 'f111e3af-1a45-42ef-9erf-4562354b8a25',
queueMgr: 'aqm',
type: 'AgentOfferConsult'
}
};

```

## eAgentWrapup

受け付けたタスクまたはコンタクトがラップアップ状態になりました。

### 例

```
Desktop.agentContact.addEventListener("eAgentWrapup", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [end\(\)](#)。

## eAgentContactHeld

受け付けたタスクまたはコンタクトが保留されました。

### 例

```
Desktop.agentContact.addEventListener("eAgentContactHeld", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [hold\(\)](#)。

## eAgentContactUnHeld

受け付けたタスクまたはコンタクトが再開されました。

### 例

```
Desktop.agentContact.addEventListener("eAgentContactUnHeld", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [unHold\(\)](#)。

## eCallRecordingStarted

受け付けたタスクまたはコンタクトのコール録音が始まりました。

### 例

```
Desktop.agentContact.addEventListener("eCallRecordingStarted", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

### 応答の例

```
const eCallRecordingStarted = {
  "data": {
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
        "ani": {
          "agentEditable": false,
          "displayName": "ani",
          "name": "ani",
          "type": "STRING",
          "value": "9997652131"
        },
        "dn": {
          "agentEditable": false,
          "displayName": "dn",
          "name": "dn",
          "type": "STRING",
          "value": "9997651073"
        },
        "ronaTimeout": {
          "agentEditable": false,
          "displayName": "ronaTimeout",
          "name": "ronaTimeout",
          "type": "STRING",
          "value": "30"
        },
        "virtualTeamName": {
          "agentEditable": false,
          "displayName": "virtualTeamName",
          "name": "virtualTeamName",
          "type": "STRING",
          "value": "qaus1-gt-requeue"
        }
      }
    },
    "callAssociatedDetails": {
      "ani": "*****",
      "dn": "*****",
      "ronaTimeout": "30",
      "virtualTeamName": "qaus1-gt-requeue"
    },
    "callFlowParams": {
      "OutdialQueue": {
        "description": "(vteam, The Outdial Queue.)",
        "name": "OutdialQueue",
        "qualifier": "vteam",
        "value": "4125",
        "valueDataType": "string"
      }
    },
    "callProcessingDetails": {
      "QMgrName": "aqm",
      "QueueId": "4125",
      "agent_ani": "9997652131",
      "ani": "*****",
      "dnis": "*****",
      "doNotRouteToAgents": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
      "outdialParkEpId": "AXRnbLmz10I4n5klYr5E",
      "outdialTransferToQueueEnabled": "true",
      "pauseDuration": "30",
      "pauseResumeEnabled": "true",
```

```

    "recordInProgress": "true",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "82",
    "virtualTeamName": "qaus1-gt-requeue",
    "vteamId": "6838",
    "vteamType": "inboundqueue"
  },
  "contactDirection": {
    "type": "OUTBOUND"
  },
  "currentVTeam": "6838",
  "interactionId": "28a8f08f-0d2f-46f9-9751-a2d9536886c1",
  "isFcManaged": false,
  "isTerminated": false,
  "media": {
    "28a8f08f-0d2f-46f9-9751-a2d9536886c1": {
      "holdTimestamp": null,
      "isHold": false,
      "mType": "mainCall",
      "mediaMgr": "vmm",
      "mediaResourceId": "28a8f08f-0d2f-46f9-9751-a2d9536886c1",
      "mediaType": "telephony",
      "participants": [
        "*****",
        "*****"
      ]
    }
  },
  "mediaChannel": "dialer",
  "mediaType": "telephony",
  "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
  "outboundType": "OUTDIAL",
  "owner": "5b24107f-aae2-45e6-9bf8-a36d18b6f0e5",
  "participants": {
    "5b24107f-aae2-45e6-9bf8-a36d18b6f0e5": {
      "channelId": "1b61e3f4-9c33-4241-9304-9ea054b88be0",
      "consultState": null,
      "consultTimestamp": null,
      "dn": "*****",
      "hasJoined": true,
      "id": "5b24107f-aae2-45e6-9bf8-a36d18b6f0e5",
      "isConsulted": false,
      "isWrapUp": false,
      "joinTimestamp": 1612429637332,
      "lastUpdated": 1612429637332,
      "name": "qaus1-gt-requeue-user qaus1-gt-requeue-user",
      "pType": "Agent",
      "queueId": "6838",
      "queueMgrId": "aqm",
      "sessionId": "56d94e8d-fbd0-4933-936d-7c19f422ec47",
      "siteId": "205",
      "teamId": "1837",
      "teamName": "integ-test-team",
      "type": "Agent",
      "wrapUpTimestamp": null
    },
    "*****": {
      "id": "9997652131",
      "pType": "Customer",
      "type": "Customer"
    },
    "9b036d89-930c-4187-a5bd-5dbb1439fe41": {
      "channelId": "3de6706f-95b7-485a-9304-30928f7ee52e",

```

```
    "consultState": null,
    "consultTimestamp": null,
    "dn": "*****",
    "hasJoined": true,
    "id": "9b036d89-930c-4187-a5bd-5dbb1439fe41",
    "isConsulted": false,
    "isWrapUp": true,
    "joinTimestamp": 1612429625705,
    "lastUpdated": 1612429636839,
    "name": "qaus1-gt-user1 qaus1-gt-user1",
    "pType": "Agent",
    "queueId": "4125",
    "queueMgrId": "aqm",
    "sessionId": "90896c91-0f77-4bc6-aec3-488a34ca88d3",
    "siteId": "205",
    "teamId": "598",
    "teamName": "integ-test-team",
    "type": "Agent",
    "wrapUpTimestamp": 1612429636839
  }
},
"previousVTeams": [
  "4125"
],
"state": "connected",
"workflowManager": null
},
"interactionId": "28a8f08f-0d2f-46f9-9751-a2d9536886c1",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"trackingId": "795e3411-191d-4524-b01c-bf672126fe02",
"type": "ContactRecordingStarted"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_9e18c618-cdd6-4bf7-be18-cafd250faabc",
"type": "RoutingMessage"
}
}
```

## eResumeRecording

受け付けたタスクまたはコンタクトの音声コールの録音を再開します。

### 例

```
Desktop.agentContact.addEventListener("eResumeRecording", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [resumeRecording\(\)](#)。

## ePauseRecording

受け付けたタスクまたはコンタクトのコール録音を一時停止します。

### 例

```
Desktop.agentContact.addEventListener("ePauseRecording", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [pauseRecording\(\)](#)。

## eConsultTransfer

受け付けたタスクまたはコンタクトの転送リクエストを打診します。

### 例

```
Desktop.agentContact.addEventListener("eConsultTransfer", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultTransfer\(\)](#)。

## eAgentblindTransferred

受け付けたタスクまたはコンタクトをエージェントまたはキューに転送します。

### 例

```
Desktop.agentContact.addEventListener("eAgentblindTransferred", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [blindTransfer\(\)](#)。

## eAgentvteamTransfer

受け付けたタスクまたはコンタクトをキューに転送します。

### 例

```
Desktop.agentContact.addEventListener("eAgentvteamTransfer", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [vTeamTransfer\(\)](#)。

## eAgentConsultCreated

受け付けたタスクまたはコンタクトの打診リクエストが作成されました。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultCreated", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consult\(\)](#)。

## eAgentConsultConferenced

受け付けたタスクまたはコンタクトの打診会議リクエストが接続されました。




---

(注) このイベントは、電子メールを除くすべてのチャンネルで会議が確立されたときに受信されます。

---

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultConferenced", (msg:
Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultConference\(\)](#)。

## eAgentConsultEnded

受け付けたタスクまたはコンタクトの打診リクエストが終了しました。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultEnded", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultEnd\(\)](#)。

## eAgentCtqCancelled

キューに対する打診リクエストが取り消されました。

### 例

```
Desktop.agentContact.addEventListener("eAgentCtqCancelled", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [cancelCtq\(\)](#)。

## eAgentConsultConferenceEnded

受け付けたタスクまたはコンタクトのコンサルト会議リクエストが終了しました。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultConferenceEnded", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultEnd\(\)](#)。

## eAgentConsulting

エージェントが着信タスクまたはコンタクトの打診を開始しました。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsulting", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultAccept\(\)](#)。

## eAgentConsultFailed

受け付けたタスクまたはコンタクトの打診リクエストは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultFailed", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consult\(\)](#)。

## eAgentConsultEndFailed

受け付けたタスクまたはコンタクトの打診を終了するリクエストは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultEndFailed", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consultEnd\(\)](#)。

## eAgentCtqFailed

キューに対する打診リクエストは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentCtqFailed", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [consult\(\)](#)

## eAgentCtqCancelFailed

キューに対する打診を取り消すリクエストは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentCtqCancelFailed", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```

応答の例の詳細については、次の項目を参照してください。 [cancelCtq\(\)](#)

## eAgentConsultConferenceEndFailed

コンサルト会議リクエストを終了するリクエストは成功しませんでした。

### 例

```
Desktop.agentContact.addEventListener("eAgentConsultConferenceEndFailed", (msg: Service.Aqm.Contact.AgentContact) => logger.info(msg));
```





## 第 10 章

# ダイヤラモジュール

Desktop.dialer モジュールのに関連する通知イベントについて、リクエストを作成し、リッスンしますアウトダイヤルアクティビティ（エージェントが開始した顧客宛ての発信コール）。

### 例

```
import {
  Desktop
} from "@wxcc-desktop/sdk";

...

const outdial = await Desktop.dialer.startOutdial({
  data: {
    entryPointId: "1212312567",
    destination: "12078995678",
    direction: "OUTBOUND",
    attributes: {},
    mediaType: "telephony",
    outboundType: "CourtesyCallback"
  }
});

logger.info("Dialer outdial: ", outdial);

// List of available subscription events:
Desktop.dialer.addEventListener("eOutdialFailed", msg => logger.info(msg));
```

メモリリークを回避するためにイベントサブスクリプションを削除するには、以下のオプションがあります。

```
// Module supports removing added listeners like:
const listener = msg => logger.info(msg);
Desktop.dialer.addEventListener("eOutdialFailed", listener);
Desktop.dialer.removeEventListener("eOutdialFailed", listener);

// Module supports one-time added listeners like:
Desktop.dialer.addOnceEventListener("eOutdialFailed", listener);
Desktop.dialer.removeOnceEventListener("eOutdialFailed", listener);

// Module supports removing all listeners like:
Desktop.dialer.removeAllEventListeners();
```

- [メソッド \(130 ページ\)](#)
- [イベント \(132 ページ\)](#)

# メソッド

## startOutdial(data)

アウトダイヤルコールを開始します。

### 例

```
await Desktop.dialer.startOutdial({
  data: {
    entryPointId: "1212312567",
    destination: "12078995678",
    direction: "OUTBOUND",
    attributes: {},
    mediaType: "telephony",
    outboundType: "CourtesyCallback"
  }
});
```

### パラメータ

名前	タイプ	説明	必須
data	オブジェクト	アウトダイヤルコールのオプション。	はい
-->entryPointId	文字列	エントリポイントの一意の識別子。	はい
-->destination	文字列	最終的な受信者のダイヤル番号。	はい
-->direction	文字列	ダイヤラから受信者へのコールの方向。	はい
-->attributes	オブジェクト	属性のオプション。	はい
-->mediaType	文字列	テレフォニー、ソーシャル、電子メール、チャットなどのメディアチャネルタイプ。	はい
-->outboundType	文字列	アウトダイヤル、サービスコールバックなどのアウトダイヤルコールのタイプ。	はい

### 戻り値

{Object} 取得されたデータを持つオブジェクト。

### 応答の例

```
const startOutdialResponse = {
  "data": {
    "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "eventType": "RoutingMessage",
    "interaction": {
      "callAssociatedData": {
```

```

    "dn": {
      "agentEditable": false,
      "displayName": "dn",
      "name": "dn",
      "type": "STRING",
      "value": "8895579172"
    },
    "ronaTimeout": {
      "agentEditable": false,
      "displayName": "ronaTimeout",
      "name": "ronaTimeout",
      "type": "STRING",
      "value": "30"
    },
    "virtualTeamName": {
      "agentEditable": false,
      "displayName": "virtualTeamName",
      "name": "virtualTeamName",
      "type": "STRING",
      "value": "Outdial Queue-1"
    }
  },
  "callAssociatedDetails": {
    "dn": "8895579172",
    "ronaTimeout": "30",
    "virtualTeamName": "Outdial Queue-1"
  },
  "callFlowParams": {
    "OutdialQueue": {
      "description": "(vteam, The Outdial Queue.)",
      "name": "OutdialQueue",
      "qualifier": "vteam",
      "value": "3264",
      "valueDataType": "string"
    }
  },
  "callProcessingDetails": {
    "QMgrName": "aqm",
    "QueueId": "3264",
    "dnis": "8895579172",
    "outdialTransferToQueueEnabled": "false",
    "ronaTimeout": "30",
    "taskToBeSelfServiced": "false",
    "tenantId": "133",
    "virtualTeamName": "Outdial Queue-1",
    "vteamId": "AXCLfZZU9S1oTdqe1OFZ"
  },
  "contactDirection": {
    "type": "OUTBOUND"
  },
  "currentVTeam": "3264",
  "interactionId": "8da312bb-5349-4d97-89d9-7bedd01b8782",
  "isFcManaged": false,
  "isTerminated": false,
  "media": {
    "8da312bb-5349-4d97-89d9-7bedd01b8782": {
      "holdTimestamp": null,
      "isHold": false,
      "mType": "mainCall",
      "mediaMgr": "vmm",
      "mediaResourceId": "8da312bb-5349-4d97-89d9-7bedd01b8782",
      "mediaType": "telephony",
      "participants": []
    }
  }
}

```

```

    },
    "mediaChannel": "dialer",
    "mediaType": "telephony",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": "OUTDIAL",
    "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "participants": {
      "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
        "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
        "consultState": null,
        "consultTimestamp": null,
        "dn": "8895579172",
        "hasJoined": false,
        "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
        "isConsulted": false,
        "isWrapUp": false,
        "joinTimestamp": null,
        "lastUpdated": 1612345959131,
        "name": "Jane Doe",
        "pType": "Agent",
        "queueId": "3264",
        "queueMgrId": "aqm",
        "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
        "siteId": "472",
        "teamId": "960",
        "teamName": "Email_Team",
        "type": "Agent",
        "wrapUpTimestamp": null
      }
    }
  },
  "previousVTeams": [],
  "state": "new",
  "workflowManager": null
},
"interactionId": "8da312bb-5349-4d97-89d9-7bedd01b8782",
"mediaResourceId": "8da312bb-5349-4d97-89d9-7bedd01b8782",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueMgr": "aqm",
"ronaTimeout": 30,
"trackingId": "cfe421df-f107-42f3-945b-a6da5dd13ccd",
"type": "AgentOfferContact"
},
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"trackingId": "notifs_a6cfbcc4-7a8c-4a2e-a9d5-6c23683cdcb7",
"type": "RoutingMessage"
}

```

## イベント

### eOutdialFailed

アウトダイヤルコールは成功しませんでした。

例

```
Desktop.dialer.addEventListener("eOutdialFailed", msg => logger.info(msg));
```

応答の例

```
const eOutdialFailed = {
  "agentId": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
  "eventType": "RoutingMessage",
  "interaction": {
    "callAssociatedData": {
      "dn": {
        "agentEditable": false,
        "displayName": "dn",
        "name": "dn",
        "type": "STRING",
        "value": "8895579172"
      },
      "ronaTimeout": {
        "agentEditable": false,
        "displayName": "ronaTimeout",
        "name": "ronaTimeout",
        "type": "STRING",
        "value": "30"
      },
      "virtualTeamName": {
        "agentEditable": false,
        "displayName": "virtualTeamName",
        "name": "virtualTeamName",
        "type": "STRING",
        "value": "Outdial Queue-1"
      }
    },
    "callAssociatedDetails": {
      "dn": "8895579172",
      "ronaTimeout": "30",
      "virtualTeamName": "Outdial Queue-1"
    },
    "callFlowParams": {
      "OutdialQueue": {
        "description": "(vteam, The Outdial Queue.)",
        "name": "OutdialQueue",
        "qualifier": "vteam",
        "value": "3264",
        "valueDataType": "string"
      }
    },
    "callProcessingDetails": {
      "QMgrName": "aqm",
      "QueueId": "3264",
      "dnis": "8895579172",
      "outdialTransferToQueueEnabled": "false",
      "ronaTimeout": "30",
      "taskToBeSelfServiced": "false",
      "tenantId": "133",
      "virtualTeamName": "Outdial Queue-1",
      "vteamId": "AXCLfZZU9S1oTdqe10FZ"
    },
    "contactDirection": {
      "type": "OUTBOUND"
    },
    "currentVTeam": "3264",
    "interactionId": "d87dac90-4f81-486c-8deb-b94bbcd21e92",
    "isFcManaged": false,
    "isTerminated": true,
    "media": {
      "d87dac90-4f81-486c-8deb-b94bbcd21e92": {
        "holdTimestamp": null,
        "isHold": false,
        "mType": "mainCall",

```

```

        "mediaMgr": "vmm",
        "mediaResourceId": "d87dac90-4f81-486c-8deb-b94bbcd21e92",
        "mediaType": "telephony",
        "participants": [
            ]
        }
    },
    "mediaChannel": "dialer",
    "mediaType": "telephony",
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "outboundType": "OUTDIAL",
    "owner": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
    "participants": {
        "7c867aa9-ec768-341a-b767-e5hd6ae7g701": {
            "channelId": "0717a2ce-76cd-4ba4-9053-71f2c78168b8",
            "consultState": null,
            "consultTimestamp": null,
            "dn": "8895579172",
            "hasJoined": false,
            "id": "7c867aa9-ec768-341a-b767-e5hd6ae7g701",
            "isConsulted": false,
            "isWrapUp": false,
            "joinTimestamp": null,
            "lastUpdated": 1612346591051,
            "name": "John Doe",
            "pType": "Agent",
            "queueId": "3264",
            "queueMgrId": "aqm",
            "sessionId": "3d017488-527a-4e89-9313-5d4eb353c789",
            "siteId": "472",
            "teamId": "960",
            "teamName": "Email_Team",
            "ttype": "Agent",
            "wrapUpTimestamp": null
        }
    },
    "previousVTeams": [
    ],
    "state": "new",
    "workflowManager": null
},
"interactionId": "d87dac90-4f81-486c-8deb-b94bbcd21e92",
"mediaType": "telephony",
"orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
"queueId": "3264",
"queueMgr": "aqm",
"reason": "CUSTOMER_UNAVAILABLE",
"reasonCode": 500,
"trackingId": "09f0d9fd-6cad-48a4-a906-be1b09a8fdc0",
"type": "AgentOutboundFailed"
}

```



## 第 11 章

# スクリーンポップモジュール

Desktop.screenpop モジュールは、スクリーンポップエンティティに関連する通知イベントについて、リクエストを作成し、リッスンします。

例：

```
Desktop.screenpop.addEventListener("eScreenPop", msg => logger.info(msg));
```

メモリリークを回避するためにイベントサブスクリプションを削除するには、以下のオプションがあります。

```
// Module supports removing added listeners like:
const listener = msg => logger.info(msg);
Desktop.screenpop.addEventListener("eScreenPop", listener);
Desktop.screenpop.removeEventListener("eScreenPop", listener);

// Module supports one-time added listeners like:
Desktop.screenpop.addOnceEventListener("eScreenPop", listener);
Desktop.screenpop.removeOnceEventListener("eScreenPop", listener);

// Module supports removing all listeners like:
Desktop.screenpop.removeAllEventListeners();
```

- [イベント \(135 ページ\)](#)

## イベント

### eScreenPop

スクリーンポップを表示するイベントをリッスンします。

例

```
Desktop.screenpop.addEventListener("eScreenPop", listener);
```

応答の例

```
const eScreenPop = {
  "data": {
    "agentId": "1736827c-10ef-4b2a-90e0-7d2a3743ded3",
    "interactionId": "4745472e-cfa2-47f3-9fcl-16359516d648",
    "queryParameters": null,
    "screenPopUrl": "https://youtube.com",
```

```
        "target": "insideDesktop",
        "type": "ScreenPop"
    },
    "orgId": "f111e3af-1a45-42ef-9erf-4562354b8a25",
    "trackingId": "notifs_90a4bf42-ba95-47a6-b8cb-c1ed105ff5ea",
    "type": "RoutingMessage"
}
```





## 第 12 章

# ショートカットキーモジュール

Desktop.shortcutKey モジュールは、ショートカットキーの操作を登録してウィジェットから呼び出します。競合を回避するには、Agent Desktopの既存のショートカットキーのリストを把握しておく必要があります。詳細については、『[Cisco Webex Contact Center Agent Desktop User Guide \(Cisco Webex Contact Center エージェントデスクトップユーザーガイド\)](#)』の「Introduction (はじめに)」の章の「Access Keyboard Shortcuts (キーボードショートカットへのアクセス)」セクションを参照してください。

例：

```
import {
  Desktop
} from "@wxcc-desktop/sdk";

console.log(Desktop.shortcutKey.DEFAULT_SHORTCUT_KEYS); //=> logs default shortcut keys
console.log(Desktop.shortcutKey.MODIFIERS); //=> logs keys modifiers

console.log(Desktop.shortcutKey.REGISTERED_KEYS); //=> logs registered keys without
modifiers

console.log(Desktop.shortcutKey.getRegisteredKeys()); //=> logs service registered keys
with modifiers (full information)

Desktop.shortcutKey.listenKeyPress((event) => {
  ...
}); //=> listen shortcuts key press

Desktop.shortcutKey.listenKeyConflict((event) => {
  ...
}); //=> listen shortcuts key conflict

Desktop.shortcutKey.listenConflictResolved(() => {}); //=> listen to shortcut key conflict
resolved status

Desktop.shortcutKey.register([
  ...
], {
  ...
}, ...]); //=> registering shortcut keys actions

Desktop.shortcutKey.unregisterKeys('widget-one-example'); //=> used to unregister on
unmount, widgetElement used for register should be provided
```

- [メソッド \(138 ページ\)](#)

# メソッド

## register()

ショートカットキーの操作を登録します。

### 例

```
Desktop.shortcutKey.register([
  {
    componentName: "Sample Comp",
    actionName: "login",
    modifierKeys: "ctrlKey_altKey",
    key: "r",
    callback: (data: Service.shortcut.EKeyInfo) => {}
  }
])
```

次の表に、ペイロードの詳細を示します。

名前	タイプ	説明	必須
componentName	文字列	機能、コンポーネント、またはウィジェットの 名前。	はい
actionName	文字列	割り当てられたショートカットキーによって実行 されるアクションまたは操作の名前。	はい
modifierKeys	文字列	修飾子キーは、ホストプラットフォームのキーボー ドショートカットで共通して使用されます。キー ボード修飾子キーの組み合わせは次のとおりです。  <ul style="list-style-type: none"> <li>• Ctrl+Shift</li> <li>• Alt+Shift</li> <li>• Ctrl+Alt</li> <li>• Shift</li> <li>• Ctrl</li> <li>• Alt</li> </ul>	はい
key	文字列	修飾子キーと組み合わせるメインキー。たとえば、 Ctrl+Shift+e では、Ctrl と Shift が修飾子キーで、e がメインキーです。	はい

名前	タイプ	説明	必須
callback	関数	クライアントコードでは、ショートカットキーの登録中にコールバック機能を追加できます。ショートカットキーのフレームワークが、キーボード keyup イベントの登録済みキーリストから一致するキーを見つけると、ショートカットフレームワークはコールバックメソッドを呼び出します。  keyup イベントは、キーが解放されると呼び出されます。	はい

### 戻り値

{Array} オブジェクトの配列。

### 応答の例

```
const register = {
  [
    {
      "widgetElement": "agentx-wc-navigation",
      "group": "Navigation",
      "action": "Open Home Page",
      "modifierKeys": "ctrlKey_altKey",
      "key": "1"
    },
    // widgetElement is the name of the web component
    // group is the name of the functionality, component, or the widget.
    {
      "widgetElement": "agentx-wc-navigation",
      "group": "Navigation",
      "action": "Open Agent Performance Statistics Page",
      "modifierKeys": "ctrlKey_altKey",
      "key": "2"
    },
    {
      "widgetElement": "agentx-wc-navigation",
      "group": "Navigation",
      "action": "Open Widget Using JS API Page",
      "modifierKeys": "ctrlKey_altKey",
      "key": "3"
    }
  ]
}
```

## getRegisteredKeys()

修飾子を使用した登録済みのショートカットキーを取得します。

### 例

```
console.log(Desktop.shortcutKey.getRegisteredKeys());
```

### 戻り値

{Array} オブジェクトの配列。

## 応答の例

```

const getRegisteredKeys = [
  ["Go to Available StatectrlKey_altKeyr", {
    "widgetElement": "agentx-react-state-selector",
    "group": "Agent State",
    "action": "Go to Available State",
    "modifierKeys": "ctrlKey_altKey",
    "key": "r"
  }],
  ["Go to Idle StatectrlKey_altKeyn", {
    "widgetElement": "agentx-react-state-selector",
    "group": "Agent State",
    "action": "Go to Idle State",
    "modifierKeys": "ctrlKey_altKey",
    "key": "n"
  }],
  ["Send EmailctrlKey_altKeys", {
    "widgetElement": "agentx-react-email-composer",
    "group": "Email Handling",
    "action": "Send Email",
    "modifierKeys": "ctrlKey_altKey",
    "key": "s"
  }],
  ["ReplyctrlKey_shiftKey6", {
    "widgetElement": "agentx-react-email-composer",
    "group": "Email Handling",
    "action": "Reply",
    "modifierKeys": "ctrlKey_shiftKey",
    "key": "6"
  }],
  ["Reply AllctrlKey_shiftKey5", {
    "widgetElement": "agentx-react-email-composer",
    "group": "Email Handling",
    "action": "Reply All",
    "modifierKeys": "ctrlKey_shiftKey",
    "key": "5"
  }],
  ["Open Outbound callctrlKey_altKeyo", {
    "widgetElement": "agentx-react-out-dial",
    "group": "Outbound",
    "action": "Open Outbound call",
    "modifierKeys": "ctrlKey_altKey",
    "key": "o"
  }],
  ["Open Notification CenterctrlKey_altKeyi", {
    "widgetElement": "wagentx-wc-menu-notification",
    "group": "Notification",
    "action": "Open Notification Center",
    "modifierKeys": "ctrlKey_altKey",
    "key": "i"
  }],
  ["Enable Silent NotificationsctrlKey_altKeyd", {
    "widgetElement": "wagentx-wc-menu-notification",
    "group": "Notification",
    "action": "Enable Silent Notifications",
    "modifierKeys": "ctrlKey_altKey",
    "key": "d"
  }],
  ["Accept Chat/Email/SocialctrlKey_altKeya", {
    "widgetElement": "agentx-react-interaction-control",
    "group": "Application",
    "action": "Accept Chat/Email/Social",
    "modifierKeys": "ctrlKey_altKey",
  }],

```

```

        "key": "a"
    }],
    [{"Switch between PopoversctrlKey_altKey", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Application",
        "action": "Switch between Popovers",
        "modifierKeys": "ctrlKey_altKey",
        "key": "p"
    }],
    [{"Expand/Collapse the PopoverctrlKey_shiftKey9", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Application",
        "action": "Expand/Collapse the Popover",
        "modifierKeys": "ctrlKey_shiftKey",
        "key": "9"
    }],
    [{"Accept all (visible) PopoversctrlKey_shiftKey4", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Application",
        "action": "Accept all (visible) Popovers",
        "modifierKeys": "ctrlKey_shiftKey",
        "key": "4"
    }],
    [{"Hold/Resume CallctrlKey_altKeyv", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "Hold/Resume Call",
        "modifierKeys": "ctrlKey_altKey",
        "key": "v"
    }],
    [{"Pause/Resume RecordingctrlKey_shiftKeyz", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "Pause/Resume Recording",
        "modifierKeys": "ctrlKey_shiftKey",
        "key": "z"
    }],
    [{"Conference Request for Call/ChatctrlKey_altKeyh", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "Conference Request for Call/Chat",
        "modifierKeys": "ctrlKey_altKey",
        "key": "h"
    }],
    [{"Consult Request for CallctrlKey_altKeyc", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "Consult Request for Call",
        "modifierKeys": "ctrlKey_altKey",
        "key": "c"
    }],
    [{"End for All ChannelsctrlKey_altKeye", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "End for All Channels",
        "modifierKeys": "ctrlKey_altKey",
        "key": "e"
    }],
    [{"Transfer Request for All ChannelsctrlKey_altKeyx", {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Interaction Control",
        "action": "Transfer Request for All Channels",
        "modifierKeys": "ctrlKey_altKey",
        "key": "x"
    }],

```

```

    }],
    [{"Save Edited CAD Variable ValuesctrlKey_altKeym", {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Save Edited CAD Variable Values",
      "modifierKeys": "ctrlKey_altKey",
      "key": "m"
    }
  ]],
    [{"Revert Edited CAD Variable ValuesctrlKey_altKeyz", {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Revert Edited CAD Variable Values",
      "modifierKeys": "ctrlKey_altKey",
      "key": "z"
    }
  ]],
    [{"Expand/CollapsectrlKey_shiftKeyy", {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Expand/Collapse",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "y"
    }
  ]],
    [{"Wrap Up ReasonctrlKey_altKeyw", {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Wrap Up Reason",
      "modifierKeys": "ctrlKey_altKey",
      "key": "w"
    }
  ]],
    [{"Open User ProfilectrlKey_altKeyu", {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Open User Profile",
      "modifierKeys": "ctrlKey_altKey",
      "key": "u"
    }
  ]],
    [{"Sign OutctrlKey_altKey1", {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Sign Out",
      "modifierKeys": "ctrlKey_altKey",
      "key": "1"
    }
  ]],
    [{"Open Keyboard Shortcuts ListctrlKey_altKeyf", {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Open Keyboard Shortcuts List",
      "modifierKeys": "ctrlKey_altKey",
      "key": "f"
    }
  ]],
    [{"Download Error ReportctrlKey_shiftKey2", {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Download Error Report",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "2"
    }
  ]],
    [{"Switch between active tasksctrlKey_shiftKey8", {
      "widgetElement": "agentx-wc-task-list",
      "group": "Task List",
      "action": "Switch between active tasks",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "8"
    }
  ]],

```

```
[{"Expand/Collapse the Task PanelctrlKey_shiftKey7", {
  "widgetElement": "agentx-wc-task-list",
  "group": "Task List",
  "action": "Expand/Collapse the Task Panel",
  "modifierKeys": "ctrlKey_shiftKey",
  "key": "7"
}},
{"Open Navigation TabctrlKey_altKeyt", {
  "widgetElement": "agentx-wc-connector",
  "group": "Connector View",
  "action": "Open Navigation Tab",
  "modifierKeys": "ctrlKey_altKey",
  "key": "t"
}},
{"RefreshctrlKey_altKeyb", {
  "widgetElement": "agentx-wc-connector",
  "group": "Connector View",
  "action": "Refresh",
  "modifierKeys": "ctrlKey_altKey",
  "key": "b"
}},
{"Open HomectrlKey_altKey1", {
  "widgetElement": "agentx-wc-navigation",
  "group": "Navigation",
  "action": "Open Home",
  "modifierKeys": "ctrlKey_altKey",
  "key": "1"
}},
{"Open Agent Performance StatisticsctrlKey_altKey2", {
  "widgetElement": "agentx-wc-navigation",
  "group": "Navigation",
  "action": "Open Agent Performance Statistics",
  "modifierKeys": "ctrlKey_altKey",
  "key": "2"
}},
{"Open Webex Experience Manager MetricsctrlKey_altKey3", {
  "widgetElement": "agentx-wc-navigation",
  "group": "Navigation",
  "action": "Open Webex Experience Manager Metrics",
  "modifierKeys": "ctrlKey_altKey",
  "key": "3"
}}
]
```

## デフォルトのショートカットキー

デフォルトのショートカットキーをログに記録します。

### 例

```
console.log(Desktop.shortcutKey.DEFAULT_SHORTCUT_KEYS);
```

### 戻り値

{Array} オブジェクトの配列。

### 応答の例

```
const DEFAULT_SHORTCUT_KEYS = {
  "agentState": [{
    "widgetElement": "agentx-react-state-selector",
    "group": "Agent State",
```

```

        "action": "Go to Available State",
        "modifierKeys": "ctrlKey_altKey",
        "key": "r"
    }, {
        "widgetElement": "agentx-react-state-selector",
        "group": "Agent State",
        "action": "Go to Idle State",
        "modifierKeys": "ctrlKey_altKey",
        "key": "n"
    }
  ]],
  "emailComposer": [
    {
      "widgetElement": "agentx-react-email-composer",
      "group": "Email Handling",
      "action": "Send Email",
      "modifierKeys": "ctrlKey_altKey",
      "key": "s"
    },
    {
      "widgetElement": "agentx-react-email-composer",
      "group": "Email Handling",
      "action": "Reply",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "6"
    },
    {
      "widgetElement": "agentx-react-email-composer",
      "group": "Email Handling",
      "action": "Reply All",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "5"
    }
  ]],
  "outDial": [
    {
      "widgetElement": "agentx-react-out-dial",
      "group": "Outbound",
      "action": "Open Outbound call",
      "modifierKeys": "ctrlKey_altKey",
      "key": "o"
    }
  ]],
  "notification": [
    {
      "widgetElement": "wagentx-wc-menu-notification",
      "group": "Notification",
      "action": "Open Notification Center",
      "modifierKeys": "ctrlKey_altKey",
      "key": "i"
    },
    {
      "widgetElement": "wagentx-wc-menu-notification",
      "group": "Notification",
      "action": "Enable Silent Notifications",
      "modifierKeys": "ctrlKey_altKey",
      "key": "d"
    }
  ]],
  "interactionPopover": [
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Application",
      "action": "Accept Chat/Email/Social",
      "modifierKeys": "ctrlKey_altKey",
      "key": "a"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Application",
      "action": "Switch between Popovers",
      "modifierKeys": "ctrlKey_altKey",
      "key": "p"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Application",

```



```

        "action": "Expand/Collapse the Popover",
        "modifierKeys": "ctrlKey_shiftKey",
        "key": "9"
    }, {
        "widgetElement": "agentx-react-interaction-control",
        "group": "Application",
        "action": "Accept all (visible) Popovers",
        "modifierKeys": "ctrlKey_shiftKey",
        "key": "4"
    }
  ]],
  "interactionControl": [
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Hold/Resume Call",
      "modifierKeys": "ctrlKey_altKey",
      "key": "v"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Pause/Resume Recording",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "z"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Conference Request for Call/Chat",
      "modifierKeys": "ctrlKey_altKey",
      "key": "h"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Consult Request for Call",
      "modifierKeys": "ctrlKey_altKey",
      "key": "c"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "End for All Channels",
      "modifierKeys": "ctrlKey_altKey",
      "key": "e"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Transfer Request for All Channels",
      "modifierKeys": "ctrlKey_altKey",
      "key": "x"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Save Edited CAD Variable Values",
      "modifierKeys": "ctrlKey_altKey",
      "key": "m"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Revert Edited CAD Variable Values",
      "modifierKeys": "ctrlKey_altKey",
      "key": "z"
    },
    {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Expand/Collapse",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "y"
    }
  ]
}

```

```

    }, {
      "widgetElement": "agentx-react-interaction-control",
      "group": "Interaction Control",
      "action": "Wrap Up Reason",
      "modifierKeys": "ctrlKey_altKey",
      "key": "w"
    }
  ],
  "userProfile": [
    {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Open User Profile",
      "modifierKeys": "ctrlKey_altKey",
      "key": "u"
    },
    {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Sign Out",
      "modifierKeys": "ctrlKey_altKey",
      "key": "l"
    },
    {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Open Keyboard Shortcuts List",
      "modifierKeys": "ctrlKey_altKey",
      "key": "f"
    },
    {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Download Error Report",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "2"
    }
  ],
  "taskList": [
    {
      "widgetElement": "agentx-wc-task-list",
      "group": "Task List",
      "action": "Switch between active tasks",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "8"
    },
    {
      "widgetElement": "agentx-wc-task-list",
      "group": "Task List",
      "action": "Expand/Collapse the Task Panel",
      "modifierKeys": "ctrlKey_shiftKey",
      "key": "7"
    }
  ],
  "connectorView": [
    {
      "widgetElement": "agentx-wc-connector",
      "group": "Connector View",
      "action": "Open Navigation Tab",
      "modifierKeys": "ctrlKey_altKey",
      "key": "t"
    },
    {
      "widgetElement": "agentx-wc-connector",
      "group": "Connector View",
      "action": "Refresh",
      "modifierKeys": "ctrlKey_altKey",
      "key": "b"
    }
  ]
}

```

## 修飾子

ショートカットキー修飾子をログに記録します。ショートカットキー修飾子は、Shift、Ctrl、Alt、またはこれらのキーの組み合わせです。

### 例

```
console.log(Desktop.shortcutKey.MODIFIERS);
```

### 戻り値

{Object} オブジェクトの応答。

### 応答の例

```
const MODIFIERS = {
  "CTRL_SHIFT": "ctrlKey_shiftKey",
  "ALT_SHIFT": "altKey_shiftKey",
  "CTRL_ALT": "ctrlKey_altKey",
  "SHIFT": "shiftKey",
  "CTRL": "ctrlKey",
  "ALT": "altKey"
}
```

## 登録済みキー

登録されたショートカットキーを、修飾子なしでログに記録します。

### 例

```
console.log(Desktop.shortcutKey.REGISTERED_KEYS);
```

### 戻り値

{Object} オブジェクトの応答。

### 応答の例

```
const REGISTERED_KEYS = {
  "EXPAND_COLLAPSE_INTERACTION_PANEL_KEY": "y",
  "SAVE_EDITED_CAD_KEY": "m",
  "REVERT_EDITED_CAD_KEY": "z",
  "HOLD_RESUME_CALL_KEY": "v",
  "TRANSFER_KEY": "x",
  "CONSULT_KEY": "c",
  "END_KEY": "e",
  "CONFERENCE_KEY": "h",
  "PAUSE_RESUME_RECORDING_KEY": "z",
  "GO_TO_AVAILABLE_KEY": "r",
  "OPEN_STATE_SELECTOR_KEY": "n",
  "SEND_EMAIL_KEY": "s",
  "REPLY_EMAIL_KEY": "6",
  "REPLY_ALL_EMAIL_KEY": "5",
  "OPEN_USER_PROFILE_KEY": "u",
  "ENABLE_SILENT_NOTIFICATION_KEY": "d",
  "OPEN_SHORTCUT_KEY_MODAL_KEY": "f",
  "DOWNLOAD_ERROR_REPORT_KEY": "2",
  "SIGNOUT_KEY": "l",
  "ACCEPT_TASK_KEY": "a",
  "SWITCH_POPOVER_KEY": "p",
}
```

## unregisterKeys()

```

    "EXPAND_COLLAPSE_POPOVER_KEY": "9",
    "OPEN_OUTDIAL_KEY": "o",
    "OPEN_WRAP_UP_KEY": "w",
    "EXPAND_COLLAPSE_TASK_LIST_PANEL_KEY": "7",
    "OPEN_NOTIFICATION_CENTER_KEY": "i",
    "OPEN_NAVIGATION_TAB_KEY": "t",
    "REFRESH_KEY": "b",
    "SWITCH_TASK_KEY": "8",
    "ACCEPT_ALL_TASK_KEY": "4"
  }
}

```

## unregisterKeys()

ショートカットキーの登録を解除します。

### 例

```

Desktop.shortcutKey.unregisterKeys([
  {
    widgetElement: "custom-element",
    group: "custom widget",
    action: "login",
    modifierKeys: "ctrlKey_altKey",
    key: "r"
  },
  {
    widgetElement: "custom-element",
    group: "custom widget",
    action: "logout",
    modifierKeys: "ctrlKey_altKey",
    key: "n"
  }
])

```

## listenKeyPress(event)

ショートカットキーの押下をリッスンします。

### 例

```

Desktop.shortcutKey.listenKeyPress(event => {
  console.log("JSSDK ShortcutKey listenKeyPress: ", event);
});

```

### パラメータ

名前	タイプ	説明	必須
event	オブジェクト	このイベントは、キーボードのキーが解放されると発生します。	はい

### 戻り値

{Object} オブジェクトの応答。

### 応答の例

```

{
  "type": "execute",

```

```

    "data": {
      "widgetElement": "agentx-react-agent-profile",
      "group": "User Profile",
      "action": "Open Keyboard Shortcuts List",
      "modifierKeys": "ctrlKey_altKey",
      "key": "f",
      "keyCombination": "Ctrl + Alt + F"
    },
    "keyboardEvent": {
      "isTrusted": true
    }
  }
}

```

## listenKeyConflict(event)

ショートカットキーの競合をリッスンします。

### 例

```

Desktop.shortcutKey.listenKeyConflict((event => {
  console.log("JSSDK ShortcutKey listenKeyPress: ", event);
}));

```

### パラメータ

名前	タイプ	説明	必須
event	オブジェクト	このイベントは、キーボードのキーが解放されると発生します。	はい

### 戻り値

{Object} オブジェクトの応答。

### 応答の例

```

{
  "type": "conflict",
  "data": {
    "widgetElement": "agentx-react-agent-profile",
    "group": "User Profile",
    "action": "Open Keyboard Shortcuts List",
    "modifierKeys": "ctrlKey_altKey",
    "key": "f",
    "isConflict": true
  }
}

```

## listenConflictResolved()

ショートカットキーの競合の解決済みステータスをリッスンします。

### 例

```

Desktop.shortcutKey.listenConflictResolved(() => {});

```

listenConflictResolved()



## 第 **III** 部

# 移行

- [Finesse ガジェットの移行 \(153 ページ\)](#)







## 第 13 章

# Finesse ガジェット の移行

- [Finesse 埋め込み Web アプリケーションガジェットの移行 \(153 ページ\)](#)
- [JavaScript ベースの Finesse ガジェットの移行 \(155 ページ\)](#)

## Finesse 埋め込み Web アプリケーションガジェットの移行

Finesse 埋め込み Web アプリケーションのガジェットを、Webex Contact Center エージェントデスクトップの iFrame ベースのウィジェットに移行できます。



(注) Finesse ガジェットの変換は、Web アプリケーションの構成、つまり iFrame に読み込めるかどうかによって異なります。

### Finesse : 埋め込み Web アプリケーションのサンプルガジェット

埋め込み Web アプリケーションのサンプルガジェットは、ガジェット内の iFrame に Web ページを表示します。これは、外部 Web アプリケーションを Finesse ガジェットに配置する例として示すことを目的としています。

埋め込み Web アプリケーションのサンプルガジェットの詳細については、<https://github.com/CiscoDevNet/finesse-sample-code/tree/master/EmbeddedWebAppSampleGadget> を参照してください。

### Webex Contact Center エージェントデスクトップ : agentx-wc-iframe

デスクトップには、iFrame 内に Web アプリケーションを読み込むための固有の機能 (Web コンポーネント) があります。デスクトップレイアウト内でコンポーネント agentx-wc-iframe を使用して、水平ヘッダー、ナビゲーションバー、補助情報ペインに Web アプリケーションを読み込むことができます。

Webex Contact Center エージェントデスクトップのウィジェットの詳細については、<https://github.com/CiscoDevNet/webex-contact-center-widget-starter/tree/master/Examples> を参照してください。

Finesse 埋め込みガジェットファイルは次のとおりです。

- EmbeddedWebApp.css
- EmbeddedWebApp.js
- EmbeddedWebApp.xml

EmbeddedWebApp.xml は、Web アプリケーションの URL を含むメインファイルです。

### 始める前に

- Finesse 埋め込みガジェットファイル EmbeddedWebApp.xml からのスクリプトタイプの詳細を理解します。

```
<script type="text/javascript">
  // initialize the gadget running the init handler defined in
  EmbeddedWebApp.js
  gadgets.HubSettings.onConnect = function () {
    finesse.modules.EmbeddedWebAppGadget.init("https://www.bing.com");
  };
</script>
```

- JSON レイアウトフォーマットを理解します。JSON レイアウトの詳細については、『[Cisco Webex Contact Center Setup and Administration Guide \(Cisco Webex Contact Center 設定およびアドミニストレーションガイド\)](#)』の「プロビジョニング (Provisioning)」の章の「デスクトップレイアウト (Desktop Layout)」セクションを参照してください。

**ステップ 1** Webex Contact Center 管理ポータルから、JSON レイアウトファイルにアクセスします。

**ステップ 2 comp** プロパティタグ内に、コンポーネント値として **agentx-wc-iframe** を入力します。

```
{
  "comp": "agentx-wc-iframe"
},
```

**ステップ 3 src attributes** プロパティタグ内に、Web アプリケーションの URL を入力します。

```
{
  "comp": "agentx-wc-iframe",
  "attributes": {
    "src": "https://www.bing.com"
  },
},
```

**ステップ 4 wrapper** プロパティタグ内に、タイトルと最大化エリア名を入力します。

```
{
  "comp": "agentx-wc-iframe",
  "attributes": {
    "src": "https://www.bing.com"
  },
  "wrapper": {
    "title": "AgentX iFrame",
    "maximizeAreaName": "app-maximize-area"
  },
},
```

**ステップ 5 iFrame** のサイズを変更するには、**style** プロパティタグを使用します。

(注) デスクトップ内の iFrame は、ネイティブの iFrame プロパティをサポートしています。

```
{
  "comp": "agentx-wc-iframe",
  "attributes": {
    "src": "https://www.bing.com"
  },
  "wrapper": {
    "title": "AgentX iFrame",
    "maximizeAreaName": "app-maximize-area"
  },
  "style": {
    "height": 504 px,
    "width": 520 px,
    "display": "inline-block",
    "align-items": "center"
  },
}
```

ステップ 6 iFrame 属性を追加するには、**attributes** プロパティタグを使用します。

```
{
  "comp": "agentx-wc-iframe",
  "attributes": {
    "src": "https://www.bing.com"
    "sandbox": "allow-scripts allow popups"
  },
  "wrapper": {
    "title": "AgentX iFrame",
    "maximizeAreaName": "app-maximize-area"
  },
  "style": {
    "height": 504 px,
    "width": 520 px,
    "display": "inline-block",
    "align-items": "center"
  },
}
```

## JavaScript ベースの Finesse ガジェットの移行

Shindig ベースのガジェットから wxcc-desktop ベースのウィジェットに直接移行する方法はありません。Shindig ガジェットと wxcc-desktop ウィジェットはテクノロジーを活用しているため、既存のソースコードをリファクタリングして wxcc-desktop ベースのウィジェットを開発することができます。Finesse ガジェットは、Apache Shindig ガジェットの仕様に基づいています。Apache Shindig の詳細については、<https://shindig.apache.org/> を参照してください。任意の JavaScript ライブラリ、フレームワーク、通常の JavaScript を使用して、UI ベースのガジェットを構築できます。最終的なコンテンツは、ガジェットの XML または HTML 内に埋め込む必要があります。

### Finesse ガジェット

静的リソース : Finesse ガジェットは、XML、HTML、CSS、JavaScript、画像などの構成要素で構成されています。これらのガジェットは Finesse 3rdpartygadget Web アプリケーションに展開されます。

要素：ガジェットの重要な要素は次のとおりです。

- <Module> タグは、ルートレベルの要素であるガジェットが XML ファイルに含まれていることを示します。
- <ModulePrefs> タグには、ガジェットに関する情報が含まれます。このタグには、タイトル、説明、作成者、その他のオプション機能と、Finesse デスクトップコンテナからのガジェット関連の構成が含まれます。
- <Content type="html"> ヘッダーは、ガジェットのコンテンツタイプが HTML であることを示します。このタグには、コンテンツ要素内に埋め込まれているすべてのマークアップコンテンツ (HTML) と静的リソースが含まれます。

### 例：サンプルガジェット

```
<Module>
  <ModulePrefs title="X-Counter" description="X-Counter Example"></ModulePrefs>
  <Content type="html">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css"/>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.29.1/moment.min.js"></script>
    <![CDATA[
  <style>
    button, p {
      display: inline-block;
    }
  </style><button aria-label="decrement">-</button><p>0</p><button
aria-label="increment">+</button><script>
  var valueElement = document.querySelector('p');
  function increment() {
    const value = valueElement.innerText;
    valueElement.innerText = parseInt(value) + 1;
  }
  function decrement() {
    const value = valueElement.innerText;
    valueElement.innerText = parseInt(value) - 1;
  }
  var incrementButton = document.querySelectorAll('button')[1];
  var decrementButton = document.querySelectorAll('button')[0];

  incrementButton.addEventListener('click', e => increment);

  decrementButton.addEventListener('click', e => decrement);
</script>
  ]]>
</Content>
</Module>
```

Cisco Finesse ガジェットの詳細については、<https://developer.cisco.com/docs/finesse/#/finesse-overview/cisco-finesse-gadgets-finesse-javascript-library-api> を参照してください。

### Webex Contact Center エージェントデスクトップのウィジェット

デスクトップのウィジェットは、ブラウザのネイティブ Web コンポーネントモジュールを活用するマイクロフロントエンドの方法に基づいています。

静的リソース：カスタムウィジェットでは、バンドルライブラリ (Webpack、Rollup、Parcel など) を使用して、すべての静的リソース (CSS、JavaScript、マークアップ、画像) を 1 つの

JavaScript ファイルにバンドルする必要があります。このバンドルは、コンテンツ配信ネットワーク (CDN) にホストする必要があります。カスタムウィジェットでは、Web コンポーネントを開発するのに任意の JavaScript フレームワークを使用できます。

デスクトップからの共有データ (ModulePrefs など) : カスタムウィジェットでは、デスクトップからの共有データまたは構成を Web コンポーネントのプロパティまたは属性として使用できます。



- (注) デスクトップの共有データまたは構成は、Finesse ガジェットの共有データまたは構成とは異なります。開発者は、wxcc-desktop で利用可能な、関連する共有データまたは構成について理解しておく必要があります。詳細については、[データプロバイダー：ウィジェットのプロパティと属性 \(10 ページ\)](#) を参照してください。

JavaScript バンドルには、Web コンポーネントとしてのルート要素が必要であり、その要素名はデスクトップレイアウト構成内に記述する必要があります。カスタムウィジェットは、任意の Web ベースのライブラリまたはフレームワーク ((例: React、Angular、Web コンポーネント) を使用して開発できます。カスタムウィジェットでは、CSS、JavaScript、マークアップ、画像などの追加の静的リソースを、Ajax またはマークアップタグ (script、link、img など) を使用して内部に読み込むことができます。

以下に、バンドルライブラリ (helloworld.js) を使用せずに通常の JavaScript で構築されたサンプルの Hello World ウィジェットを例として示します。この例では、ウィジェット名は <x-counter> です。

#### 例：バンドルライブラリを使用しないサンプルのウィジェット

```
const template = document.createElement('template');
template.innerHTML = `
<style>
  button, p {
    display: inline-block;
  }
</style>
<button aria-label="decrement">-</button>
<p>0</p>
<button aria-label="increment">+</button>
`;

class XCounter extends HTMLElement {

  increment() {
    const value = this.valueElement.innerText;
    this.valueElement.innerText = parseInt(value) + 1;
  }

  decrement() {
    const value = valueElement.innerText;
    valueElement.innerText = parseInt(value) - 1;
  }

  constructor() {
    super();

    this.attachShadow({
      mode: 'open'
    });
  }
}
```

```
});  
this.shadowRoot.appendChild(template.content.cloneNode(true));  
  
this.valueElement = this.shadowRoot.querySelector('p');  
this.incrementButton = this.shadowRoot.querySelectorAll('button')[1];  
this.decrementButton = this.shadowRoot.querySelectorAll('button')[0];  
  
this.incrementButton.addEventListener('click', e => this.increment);  
  
this.decrementButton.addEventListener('click', e => this.decrement);  
}  
}  
  
customElements.define('x-counter', XCounter);
```

## 翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。