Amazon Connect Streams API 解説
Amazon Connect 再入門シリーズ

AWS Black Belt Online Seminar

卯月 翔平

Amazon Connect Specialist SA
2023/1

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 動画を一時停止・スキップすることで、興味がある分野・項目だけの聴講も可能、スキマ時間の学習にもお役立ていただけます
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/
  - https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY

内容についての注意点

- 本資料では2023年1月時点のサービス内容および価格についてご説明しています。最新の情報は AWS 公式ウェブサイト(https://aws.amazon.com/)にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます

自己紹介

名前：卯月 翔平（うづき しょうへい）

所属：アマゾン ウェブ サービス ジャパン合同会社
プロダクティビティ アプリケーションズ事業本部
Amazon Connect Specialist SA

経歴：前職では IoT デバイスを管理する Web サービスの開発に従事

好きな AWS サービス：Amazon Connect, AWS Lambda

本セミナーの目的

- Amazon Connect Streams API の概要、使用方法をご理解いただくこと
- また、実際にどのように応用できるかをご理解いただくこと

本セミナーの対象者

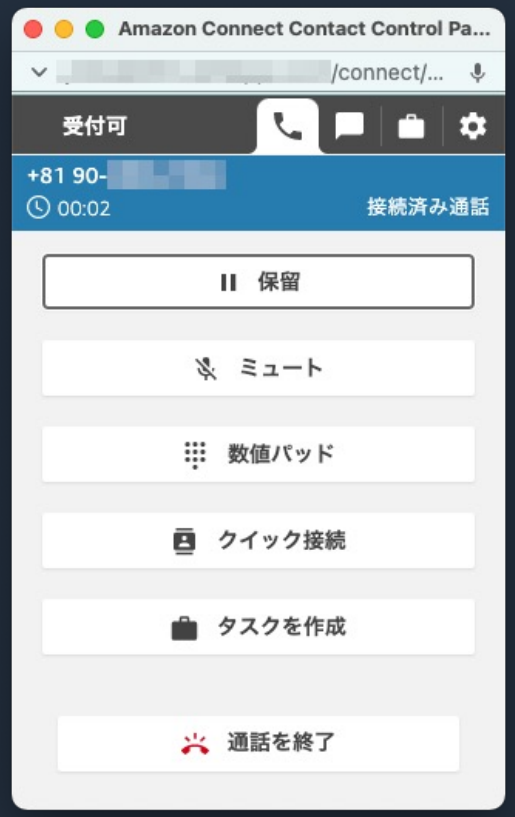
- Amazon Connect の利用方法を理解している方
- Amazon Connect のソフトフォンと Web アプリケーションの連携方法を知りたい方
- HTML や JavaScript の基本的な使い方を知っている方

アジェンダ

1. Amazon Connect Streams API とは
2. Amazon Connect Streams API の使い方
3. まとめ

Amazon Connect Streams API とは

Amazon Connect のソフトフォン



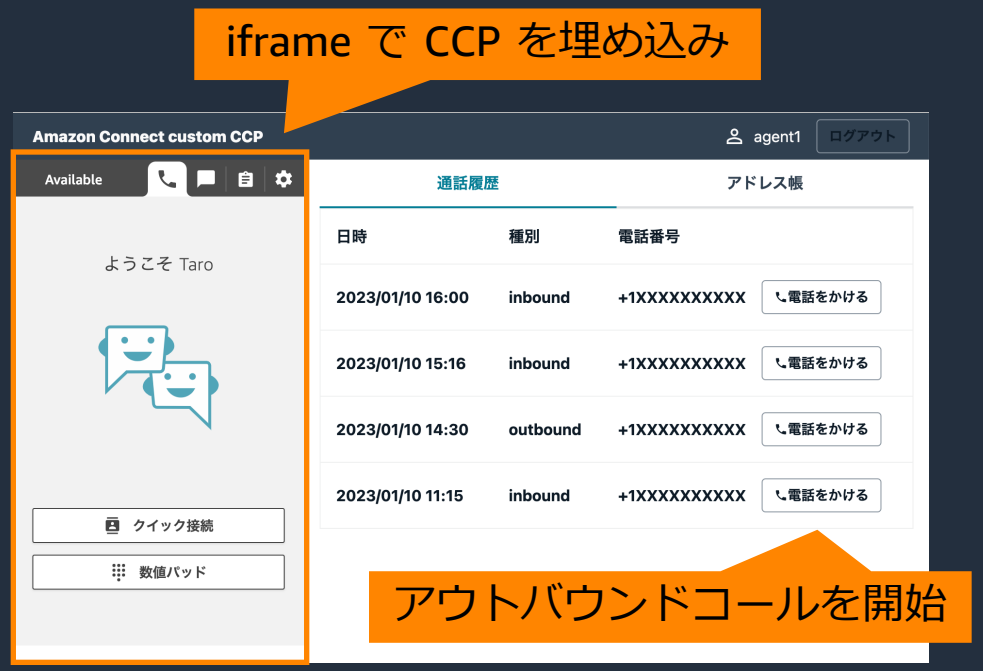
ブラウザベースのソフトフォン
(Contact Control Panel = CCP)

システム要件

- ブラウザ
  - Google Chrome / Mozilla Firefox / Microsoft Edge
  - 上記いずれかの最新 3 バージョン
- ネットワーク
  - 接続されたワークステーションあたり 100 Kbps の帯域幅
- PC
  - メモリ 2 GB、プロセッサ 2 GHz
- ヘッドセット
  - USB ヘッドセットを推奨

CCP を使用する際のエージェントのヘッドセットならびにワークステーションの要件
https://docs.aws.amazon.com/ja_jp/connect/latest/adminguide/ccp-agent-hardware.html

Amazon Connect Streams API の特徴

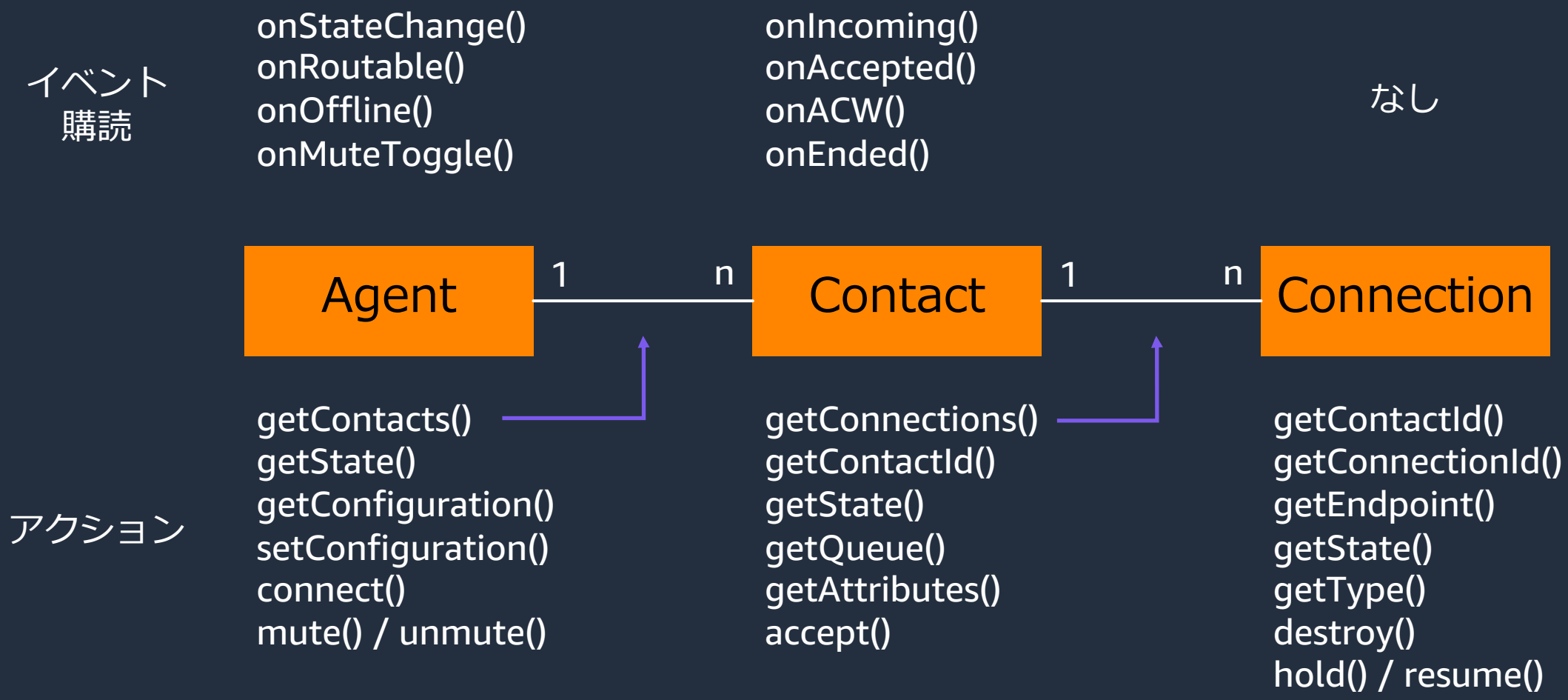


Streams API を使ったサンプル画面

- JavaScript から利用可能
- CCP を iframe でウェブページに埋め込む
  - この iframe を非表示にすることで、
    完全に独自の UI を作り込むことも可能
- コールバック形式で各種イベントを受け取る
  - イベント例: エージェントの初期化、新規コンタクト
  - 受け取ったオブジェクト経由で各種操作が可能
    - 例: アウトバウンドコールの開始
    - 例: ミュート、ミュート解除
    - 例: コンタクトを受信、コンタクトを拒否

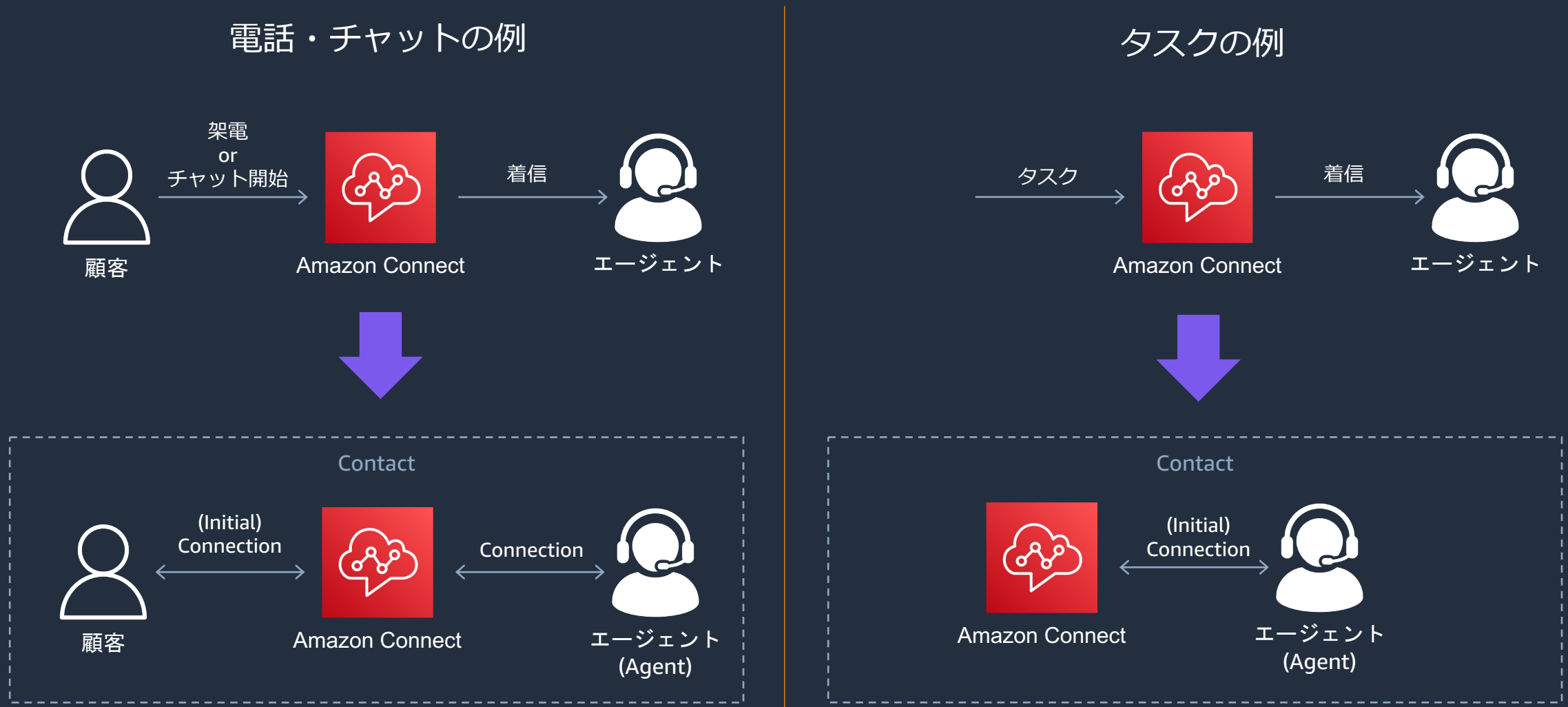
https://github.com/amazon-connect/amazon-connect-streams

Amazon Connect Streams API 全体像



https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md から抜粋

Amazon Connect Streams API 対応関係



Agent API

Streams API をロードするとグローバル変数として connect が使用可能になる

- 取得方法
  - connect.agent() のコールバック関数が初期化完了時に呼ばれ、その引数として取得可能
  - 初期化完了後であれば new connect.Agent() で取得可能
- エージェントとしてイベントの購読やアクションを行うことが可能

メソッド	説明
onStateChange()	ステータス変更時
onRoutable()	Available に変化時
onAfterCallWork()	ACW に変化時
onOffline()	Offline に変化時
onError()	エラー発生時
onMuteToggle()	ミュート状態の変化時
onSpeakerDeviceChanged()	スピーカー設定の変化時

メソッド	説明
getContacts()	現在のコンタクト一覧を取得
getState() / setState()	ステータスの取得、設定
getRoutingProfile()	ルーティングプロファイルを取得
getName()	エージェントの名前を取得
connect()	アウトバウンドコールの開始
mute() / unmute()	ミュート、ミュート解除
setSpeakerDevice()	音声出力するスピーカーを設定

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api から抜粋

Contact API

- 取得方法
  - connect.contact() のコールバック関数が初期化完了時に呼ばれ、その引数として取得可能
  - Agent API の getContacts() でも取得可能
- コンタクトとしてイベントの購読やアクションを行うことが可能

メソッド	説明
onConnecting()	着信時(リング中、応答前)
onAccepted()	応答時
onConnected()	通話確立時
onMissed()	不応答時
onEnded()	通話終了時
onError()	エラー時
addConnection()	通話に第三者を追加
conferenceConnections()	接続中のコネクションで会議を開始

メソッド	説明
getConnections()	紐づいているコネクション一覧を取得
getContactId()	コンタクト ID を取得
getType()	コンタクトの種類を取得
getState()	コンタクトの状態を取得
getQueue()	コンタクトのキューを取得
getAttributes()	コンタクト属性を取得
accept() / reject()	通話の受信、拒否

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api から抜粋

Connection API

- 取得方法
  - Contact API の getConnections(), getInitialConnection() などから取得可能
- コネクションの種類 (Voice, Chat, Task) によってメソッドに差分がある
  - Voice の場合は isMute() や muteParticipant() などが追加される

メソッド	説明
getContactId()	紐づいているコンタクト ID を取得
getConnectionId()	コネクション ID を取得
getEndpoint()	電話番号などを含むエンドポイントを取得
getState()	コネクションの状態を取得
getType()	コネクションの種類を取得
destroy()	コネクションを破棄
sendDigits()	DTMF の情報を送信

メソッド	説明
isInitialConnection()	最初のコネクションかどうか
isActive()	アクティブかどうか
isConnected()	接続されているか
isConnecting()	着信の途中かどうか
isOnHold()	保留されているか
hold() / resume()	保留、保留解除

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#connection-api から抜粋

Amazon Connect Streams API の使い方

事前準備: ウェブサイト構築

Amazon Connect Streams API を使用するウェブサイトの準備

- Amazon S3 + Amazon CloudFront
  - S3 に HTML や JavaScript のファイルを置き、CloudFront から HTTPS で配信
  - CloudFront を使用して Amazon S3 バケットに対する HTTPS リクエストを処理する (*1)
- AWS Amplify hosting
  - amplify の cli から S3 へのアップロードや CloudFront の設定が行われる
  - AWS Amplify Hosting - User Guide (*2)
  - Hosting - Overview - AWS Amplify Docs (*3)

注: 上記の構築方法は例となります。また、既存のウェブサイトがある際は、そちらを使っていただいても問題ありません。

*1 https://aws.amazon.com/jp/premiumsupport/knowledge-center/cloudfront-https-requests-s3/
*2 https://docs.aws.amazon.com/amplify/latest/userguide/welcome.html
*3 https://docs.amplify.aws/cli/hosting/hosting/

事前準備: 承認済みオリジンへの追加

1. AWS のマネジメントコンソールにログイン
2. Amazon Connect のコンソールを開く
3. 使用予定のインスタンスを選択
4. サイドメニューから承認済みオリジンを選択
5. ウェブアプリとして使用する URL を追加

https://docs.aws.amazon.com/ja_jp/connect/latest/adminguide/app-integration.html

Amazon Connect Streams API の読み込み

JavaScript ファイルの参照

アップデートは随時行われるので追従が必要

- npm から入手
  - webpack, vite などのバンドラーを利用する際は import, require から利用可能
  - jsDelivr などの CDN を利用することで直接ウェブページから参照することも可能
    - 後述の最小構成ではこの方式を採用
- GitHub から入手
  - release/connect-streams.min.js などをコピーして利用可能

```
import "amazon-connect-streams";
```

or

```
<script type="text/javascript" src="connect-streams-min.js"></script>
```

Amazon Connect Streams API の読み込み

Streams API の初期化

ページのロードが終わった後に実行される

CCP はこの div のサイズで表示される

```
<body onload="init()">
  <div id="container-div" style="width: 400px; height: 600px"></div>
  <script type="text/javascript">
    const containerDiv = document.getElementById("container-div");
    const instanceURL = "https://your-instance-alias.my.connect.aws/ccp-v2/";

    function init() {
      connect.core.initCCP(containerDiv, {
        ccpUrl: instanceURL,
        loginPopup: true,
        loginOptions: {
          autoClose: true,
        },
        softphone: {
          allowFramedSoftphone: true,
        },
      });
    }
  </script>
</body>
```

古いインスタンスの場合形式が異なります(*1)

この例でのパラメータは最小限になっているため、
実際に利用する際はドキュメント(*2)をご確認ください

*1 https://docs.aws.amazon.com/ja_jp/connect/latest/adminguide/amazon-connect-release-notes.html#new-domain

*2 https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#connectcoreinitccp

Amazon Connect Streams API の読み込み

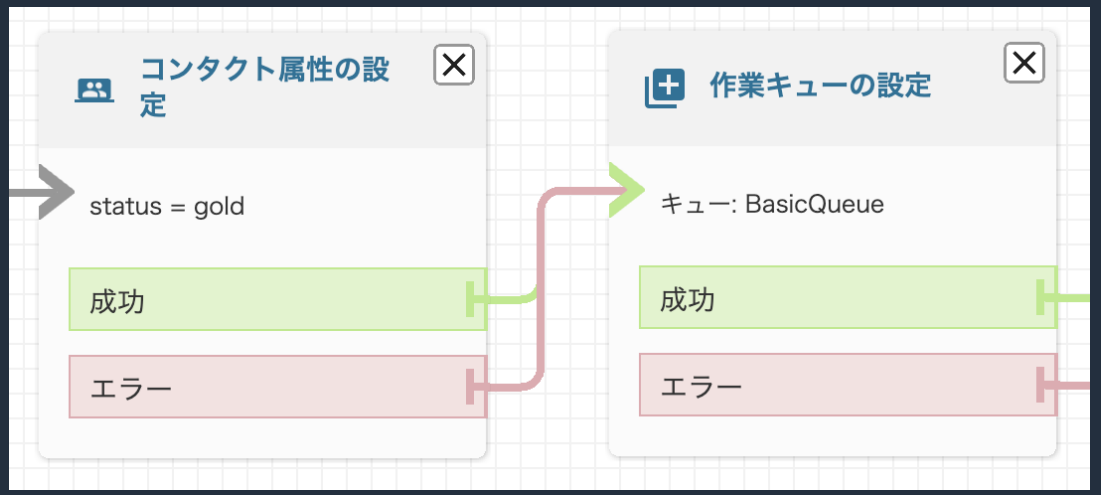
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <script type="text/javascript”
      src="https://cdn.jsdelivr.net/npm/amazon-connect-streams@2.4.4/release/connect-streams-min.js">
    </script>
  </head>
  <body onload="init()">
    <div id="container-div" style="width: 400px; height: 600px;"></div>
    <script type="text/javascript">
      const containerDiv = document.getElementById("container-div");
      const instanceURL = "https://your-instance-alias.my.connect.aws/ccp-v2/";

      function init() {
        connect.core.initCCP(containerDiv, {
          ccpUrl: instanceURL,
          loginPopup: true,
          loginOptions: {
            autoClose: true,
          },
          softphone: {
            allowFramedSoftphone: true,
          },
        });
      }
    </script>
  </body>
</html>
```

実際に動作する最小構成

Amazon Connect Streams API の利用法

イベントの購読: コンタクト(属性の取得)

- 新規コンタクトがルーティングされてきた場合に connect.contact() のコールバック関数が呼ばれる
- 左下に示すようなコンタクトフローを経由した場合、右下のサンプルコードの変数は次のようになる
  - queueName: BasicQueue
  - status: gold




```
connect.contact((contact) => {
  const queueName = contact.getQueue().name;
  const attributes = contact.getAttributes();
  const status = attributes.status.value;

  // BasicQueue, gold
  console.log(queueName, status);
});
```

このサンプルではコンソールにログを出力するのみだが、実際はこの値を使ってキューの名前や会員ステータスをエージェントにわかりやすく表示することができる

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api

Amazon Connect Streams API の利用法

イベントの購読: コンタクト(電話番号取得)

- 音声以外のチャネルでは電話番号は関係ないため、コンタクトの種類をチェック
  - ここでは VOICE だけを見ているが、 QUEUE_CALLBACK の場合も電話番号を取得可能
- 取得した電話番号からお客さま情報を取得して、画面の表示に活用可能

```
connect.contact((contact) => {
  if (contact.getType() !== connect.ContactType.VOICE) {
    return;
  }
  const connection = contact.getInitialConnection();
  const endpoint = connection.getEndpoint();

  console.log(endpoint.phoneNumber);
});
```

通話を開始したコネクションを取得

E.164 形式

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api

Amazon Connect Streams API の利用法

イベントの購読: コンタクト(待ち時間の取得)

- お客さまがキューに入った状態で待っている時間を確認する
  - contact.getQueueTimestamp() でキューにコンタクトが入った時刻が取得可能
- JavaScript の Date の getTime() はミリ秒を返す

```
connect.contact((contact) => {
  const timestamp = contact.getQueueTimestamp();
  const now = new Date();

  const waitingMillisecs = now.getTime() - timestamp.getTime();
  const waitingSeconds = Math.floor(waitingMillisecs / 1000);

  console.log(waitingSeconds);
});
```

お客様をお待たせしているかの判断に利用可能

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api

Amazon Connect Streams API の利用法

JavaScript から CCP を操作する：コンタクトの受信、破棄

- contact.onAccepted() を利用して、
  受信してから 10 秒たったコネクションを破棄するコールバック関数を登録
- コールバック関数の登録後、contact.accept() を利用して自動でコンタクトを受信

```
connect.contact((contact) => {
  contact.onAccepted(() => {
    setTimeout(() => {
      contact.getAgentConnection().destroy({
        success: () => console.log("destroy: success"),
        failure: (err) => console.log("destroy: failure", err),
      });
    }, 10 * 1000);
  });

  contact.accept();
});
```

destroy() が成功したら success、失敗したら failure が呼ばれる

説明のために用意したサンプルコードのため実際にこのような実装をすることはないが、
JavaScript 側から CCP を操作できることを確認できる。

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#contact-api

Amazon Connect Streams API の利用法

JavaScript から CCP を操作する：アウトバウンドコールの開始

- agent.connect() メソッドにエンドポイントを指定すると CCP からアウトバウンドコールが開始される
  - queueARN を指定することで特定のキューに紐づく発信者 ID 番号を指定可能
  - 多くの発信者 ID 番号を使い分けたい場合は、アウトバウンドウィスパーフローから「電話番号を呼び出す」ブロックを利用して動的に変更するのを推奨(*1)
- agent.getAllQueueARNs() はルーティングプロファイルに紐づいているキューの ARN の一覧が取得可能

E.164 形式

```
const endpoint = connect.Endpoint.byPhoneNumber("+8180YYYYZZZZ");
const agent = new connect.Agent();
agent.connect(endpoint, {
  queueARN: agent.getAllQueueARNs()[0],
  success: () => console.log("connect: success"),
  failure: (err) => console.log("connect: failure", err),
});
```

簡単のため最初の ARN で決め打ち
この ARN のキューには発信者 ID 番号を紐づける

*1 https://docs.aws.amazon.com/ja_jp/connect/latest/adminguide/queues-callerid.html#using-dynamic-caller-id
https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#agent-api

Amazon Connect Streams API の利用法

JavaScript から CCP を操作する：エージェントのミュート

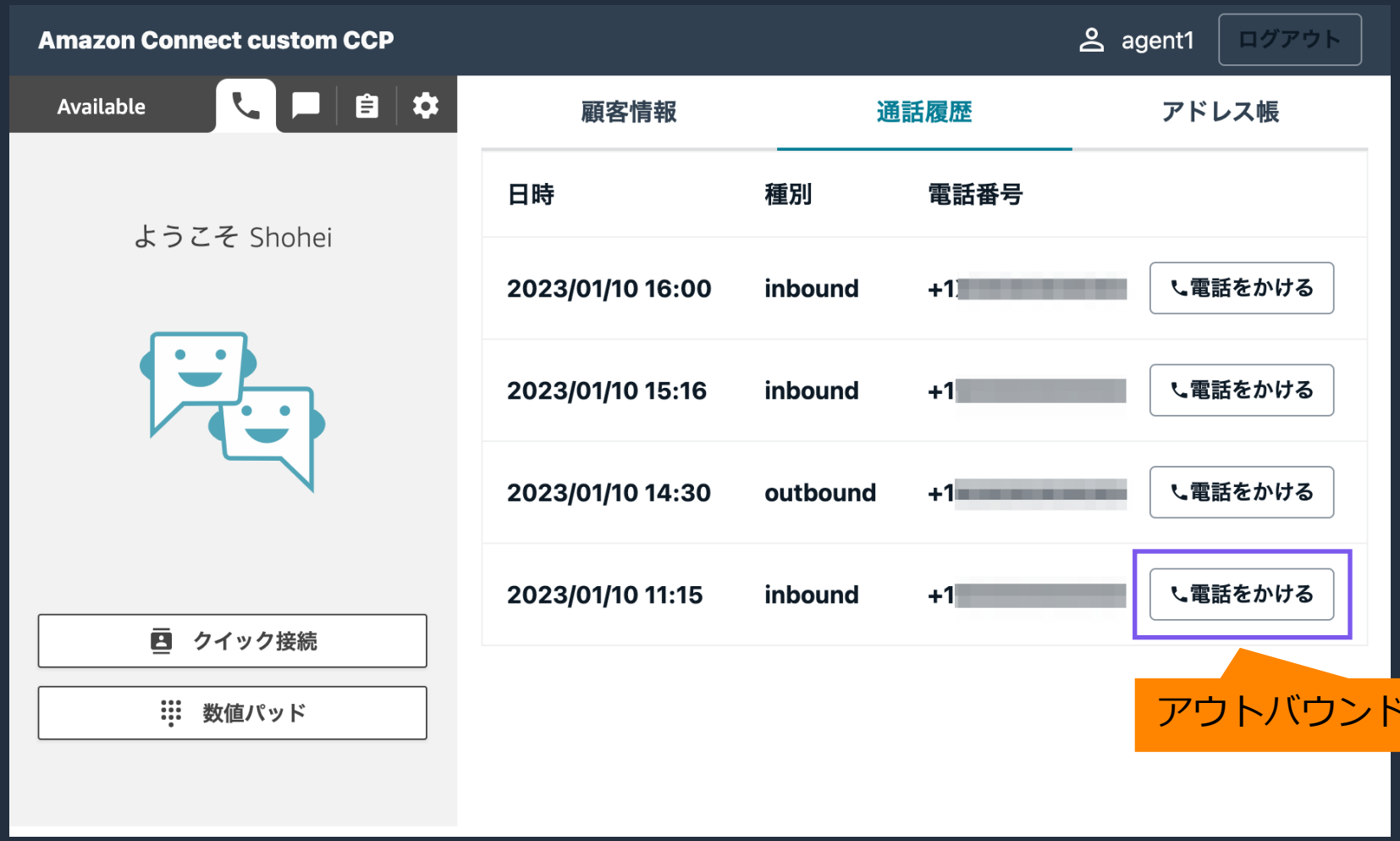
- コンタクトを受信した際に、自動でエージェントをミュートにする例
  - デモの際に音声のハウリングなどを防ぎたい場合に利用可能

```
connect.contact((contact) => {
  contact.onAccepted(() => {
    const agent = new connect.Agent();
    agent.mute();
  });
});
```

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#agent-api

Amazon Connect Streams API の利用法: カスタム CCP の構成例

Amazon Connect Streams API の利用法: カスタム CCP の構成例



Amazon Connect Streams API の利用法

Streams API から Agent Workspace の機能を利用する

- connect.core.initCCP() ではなく connect.agentApp.initApp() で初期化を行う
- 現時点では、Customer Profiles、Wisdom、Step-by-step guides の UI を利用可能
- それぞれの機能を別々に初期化できるため、必要な部分のみを組み込み可能
- Step-by-step guides の機能は初期化に加えて、表示する内容を制御するコンタクトフローを指定する必要があるため、下記のドキュメントを参考に連携するコードを書いておく

注: 2023年1月時点では Step-by-step guides はプレビュー機能であり、次のリージョンでご利用いただけます。
US East (N. Virginia), US West (Oregon), Europe (London), Asia Pacific (Sydney)

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#connectagentappinitappname-containerid-appurl-config

Amazon Connect Streams API の利用法

Streams API 単体では提供されていない機能

- Amazon Connect Streams API を用いてチャットやタスク、あるいは WebRTC 周りを
  カスタマイズしたい場合は追加で以下のライブラリを読み込む必要がある
  - https://github.com/amazon-connect/amazon-connect-chatjs
    - Amazon Connect Participant Service に依存しており AWS SDK を利用する場合は注意が必要
  - https://github.com/amazon-connect/amazon-connect-taskjs
  - https://github.com/aws/connect-rtc-js
    - CCP が提供する標準の機能を使わずに Amazon Connect が提供する API を直接使用し、
      WebRTC などの低レイヤーのカスタマイズ性を提供する

https://github.com/amazon-connect/amazon-connect-streams/blob/master/Documentation.md#connectcoreinitccp

まとめ

Amazon Connect Streams API 入門まとめ

- Amazon Connect Streams API を利用することで
  Web アプリケーションに Amazon Connect との連携を実装可能
- コンタクトフローの中で設定したコンタクト属性や電話番号などを
  JavaScript から取得でき、画面上に表示可能
- JavaScript 側から Streams API を利用して着信の応答や切断、
  アウトバウンドコールの開始など CCP を制御することが可能

Amazon Connect の学習リソース

Amazon Connect 関連の学習リソース

- AWS Hands-on for Beginners - Amazon Connectによる基本的なコンタクトセンター構築
- Amazon Connect 管理者ガイド
- AWS Black Belt Online Seminar - Amazon Connect シリーズ
- AWS WorkShop - Amazon Connect シリーズ
- AWS Contact Center Blog
- AWS What’s new - カスタマーエンゲージメント

Amazon Connect 再入門 シリーズの紹介

AWS Black Belt Online Seminar
Amazon Connect 再入門 シリーズ
シリーズの紹介

タイトル（仮）	概要
コンタクトフロー解説	デフォルトのコンタクトフローを中心に機能解説
フローブロック紹介	コンタクトフローを構成する機能ブロックの仕様、利用シーン、ユースケースを紹介
コンタクトルーティング徹底解説	Amazon Connect での高度なルーティングメソッドについての解説
エージェント機能解説	CCP のオペレーション、複数チャネルの利用、エージェントワークスペース
スーパーバイザー機能の紹介	• リアルタイムメトリクス、履歴メトリクスの使い方 • コンタクトの検索 (CTR) による高度な分析 • ダッシュボード
セキュリティのベストプラクティス	ID 管理、データ暗号化、通信経路、セキュリティプロファイル、コンプライアンス対応など

本資料に関するお問い合わせ・ご感想

技術的な内容に関しましては、有料の AWS サポート窓口へ
お問い合わせください

https://aws.amazon.com/jp/premiumsupport/

料金面でのお問い合わせに関しましては、カスタマーサポート窓口へ
お問い合わせください（マネジメントコンソールへのログインが必要です）

https://console.aws.amazon.com/support/home#/case/create?issueType=customer-service

具体的な案件に対する構成相談は、後述する個別相談会をご活用ください

ご感想は Twitter へ！ハッシュタグは以下をご利用ください
#awsblackbelt

その他コンテンツのご紹介

ウェビナーなど、AWS のイベントスケジュールをご参照いただけます

https://aws.amazon.com/jp/events/

ハンズオンコンテンツ

https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/

AWS 個別相談会

AWS のソリューションアーキテクトと直接会話いただけます

https://pages.awscloud.com/JAPAN-event-SP-Weekly-Sales-Consulting-Seminar-2021-reg-event.html

Thank you!