

Kompira Pigeon 基本マニュアル

目次

1.	はじめに	3
1.1.	Pigeon とは	3
1.2.	Pigeon 導入のメリット	4
2.	用語.....	4
3.	アクセス方法	5
4.	機能とはたらき	5
4.1.	連絡先.....	6
4.1.1.	連絡先とは.....	6
4.1.2.	連絡先管理画面.....	6
4.2.	コールフロー	7
4.2.1.	コールフローとは	7
4.2.2.	コールフロー管理画面	7
4.2.3.	最大ループ回数とは.....	8
4.3.	ガイドダンス	8
4.3.1.	ガイドダンス管理画面.....	8
4.3.2.	応答とは	9
4.3.3.	ガイドダンス内のパラメータ展開.....	10
4.4.	連絡履歴	11
4.4.1.	連絡履歴確認画面	11
4.5.	ツール	13
4.5.1.	ツール利用画面.....	13
4.5.2.	コマンドジェネレータとは.....	13
5.	設定の流れ.....	15
5.1.	連絡先の作成.....	15
5.2.	コールフローの作成	18
5.3.	ガイドダンスの作成.....	20
5.4.	テスト実行	23
5.5.	テスト結果の確認.....	27
5.6.	API トークンの準備	29
5.7.	API 実行コマンドの生成.....	29
5.8.	外部からの呼び出し	31
6.	ガイドダンス事例紹介	32

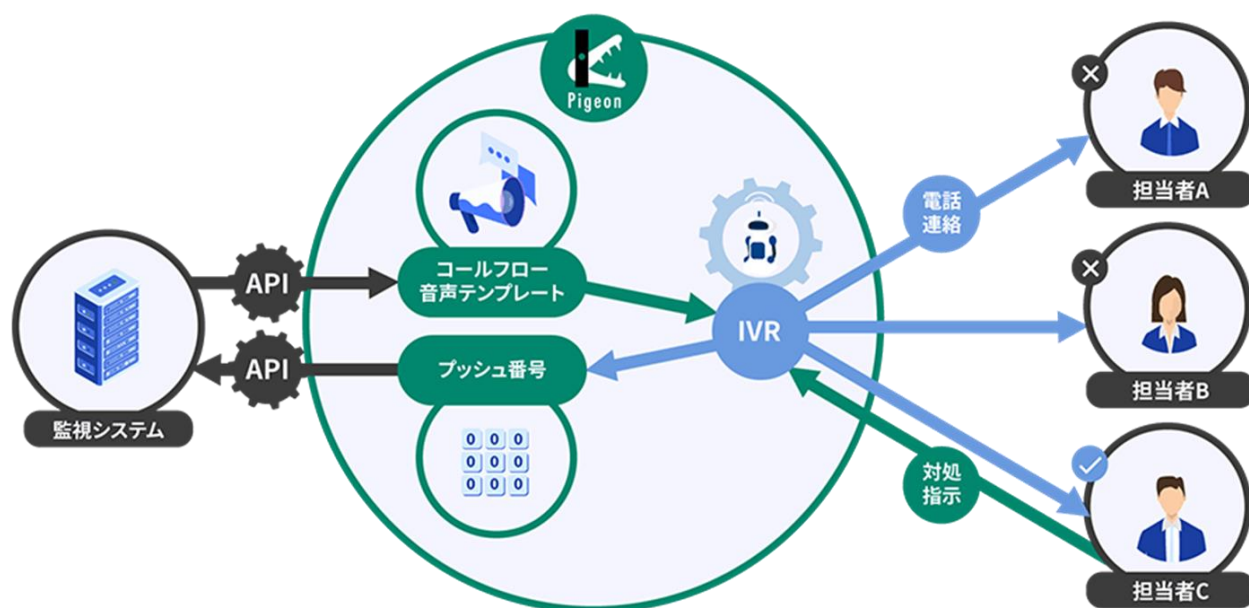
6.1.	事例 ①…使いまわしが容易なガイドンスを作成する	32
6.2.	事例 ②…電話終了のパターンを複数準備する	33
7.	補足資料	34
7.1.	API について	34
7.2.	API ドキュメント	34
7.3.	API 実行とコマンド構文例	35
7.4.	オブジェクト ID について	36

1. はじめに

1.1. Pigeon とは

Kompira Pigeon（以下 Pigeon）は、プッシュボタン応答に対応した自動電話/読み上げサービスです。
簡単設定でエスカレーション電話を自動化します。

万が一、一人の担当者が気づけなかったとしても、次の担当者に連絡し、対応を確認するため、重大なアラートに担当者が気付けないリスクを軽減します。



1.2. Pigeon 導入のメリット

電話連絡のフローをサービスごとに登録できます。

Pigeon では、電話連絡のフローを自由に作成し、架電順、電話番号、ループ回数などをフローごとに設定できるようになっています。作成フロー数に制限はないため、サービスや現象ごとにフローの細かい設定が可能です。

番号ごとに挙動の割り当てができます。

電話連絡中のプッシュボタンに対する挙動の割り当てができます。例えば 1 番には「連絡の終了」、2 番には「連絡の継続」、3 番には「自分が離脱する」など、個別に挙動の割り当て設定が可能です。

複数のボタンに「連絡の終了」を割り当て可能で、ツール連携において効果を発揮します。

誰に繋がって誰が対応をしたのか。履歴が確認できます。

電話連絡開始から終了までの一連の履歴を記録できます。どのようなインシデントに、誰がどのように対応したのかをはじめ、「電話連絡開始日時」「ループ回数」「電話連絡に対する各担当者の反応」「最終的に押されたプッシュボタン」など、一連の情報が記録されるため、対応の状況が簡単に把握できます。

2. 用語

本ドキュメントに登場する用語について説明します。

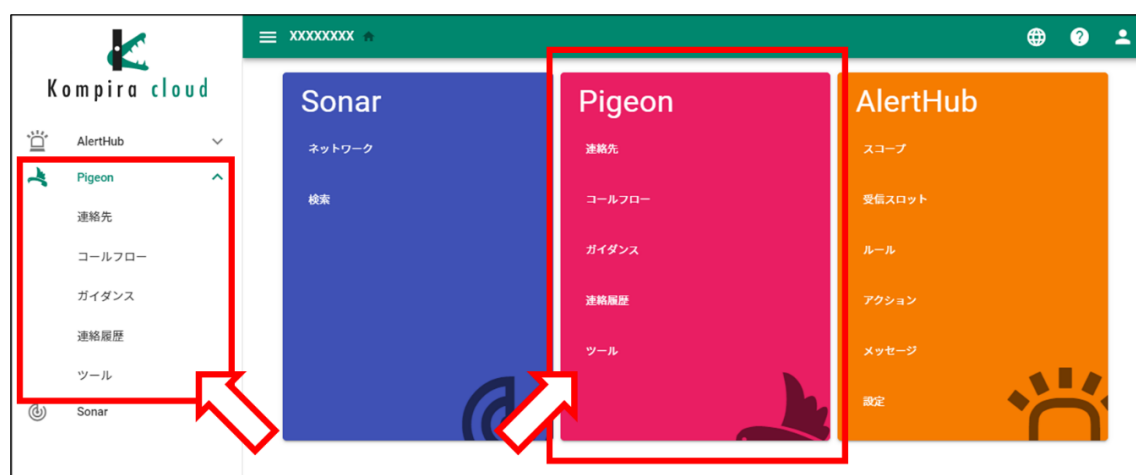
用語	説明
WEBGUI	本ドキュメントでは、Kompira cloud の WEBGUI を指す
Pigeon	Kompira Pigeon の略称
連絡先 (ID)	電話連絡時の連絡先電話番号を管理するオブジェクトとその ID
コールフロー(ID)	電話連絡時の連絡順やループ回数を管理するオブジェクトとその ID
ガイダンス(ID)	電話連絡時に再生するメッセージや応答を管理するオブジェクトとその ID
応答	電話連絡時に利用するガイダンスに含まれる定義情報。 押されたプッシュボタンに対しての挙動や再生するメッセージを定義する。
mustache 記法	様々な言語で使えるテンプレートエンジン mustache に対応した記法。 Pigeon では、ガイダンスにおいて「 <code>{{ }}</code> 」内に記載されたパラメータの展開が可能。
API	「Application Programming Interface」の略で、異なるプログラム同士が情報のやりとりを行う仕組み
API トークン	API を実行する際に必要となる認証情報。WEBGUI にて発行/管理を行う。
JSON	「JavaScript Object Notation」の略で、JavaScript の中でデータを簡単に表現するための書式。文字列・数値・配列・オブジェクトなどを表現できる。
curl	様々なプロトコルを利用し、対象の URL と通信を行う為のツールまたはコマンド

3. アクセス方法

WEBGUI より、以下の方法にて Pigeon の各機能にアクセスを行うことが可能です。

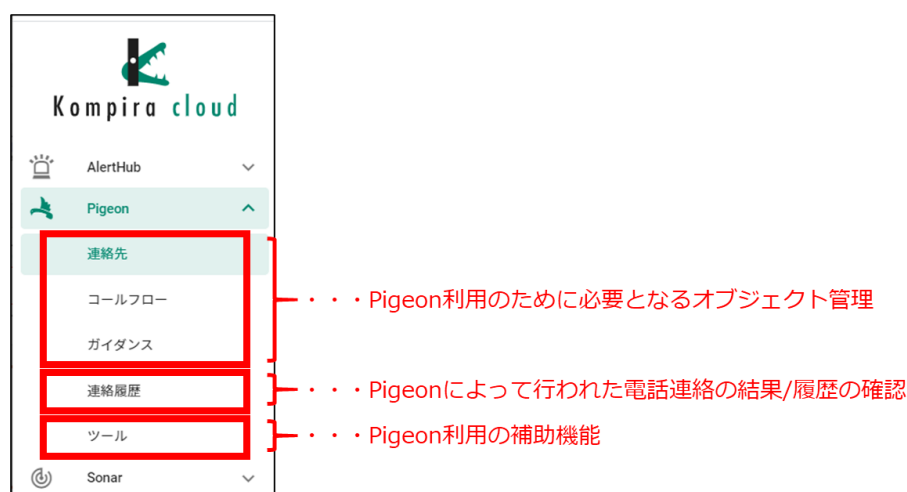
- ・WEBGUI トップページにて右側「Pigeon」領域内の各リンクをクリック
- ・WEBGUI 左側メニュー領域にて「Pigeon」をドリルダウンし各リンクをクリック
(WEBGUI トップページ以外からもアクセス可能。)

前者はトップページでしか利用できませんが、後者は WEBGUI のどのページからでも利用可能です。



4. 機能とはたらき

Pigeon の機能は、Pigeon を利用するために必要な「連絡先」「コールフロー」「ガイダンス」と呼ばれるオブジェクトの管理機能と、電話連絡の結果を確認するための「連絡履歴」、および Pigeon を利用するための補助機能となる「ツール」にて構成されています。



はじめに、それぞれのオブジェクトや機能について説明していきます。

4.1. 連絡先

4.1.1. 連絡先とは

「連絡先」とは、Pigeon が電話連絡を行う担当者の情報を管理するオブジェクトです。

以下の要素で構成されています。

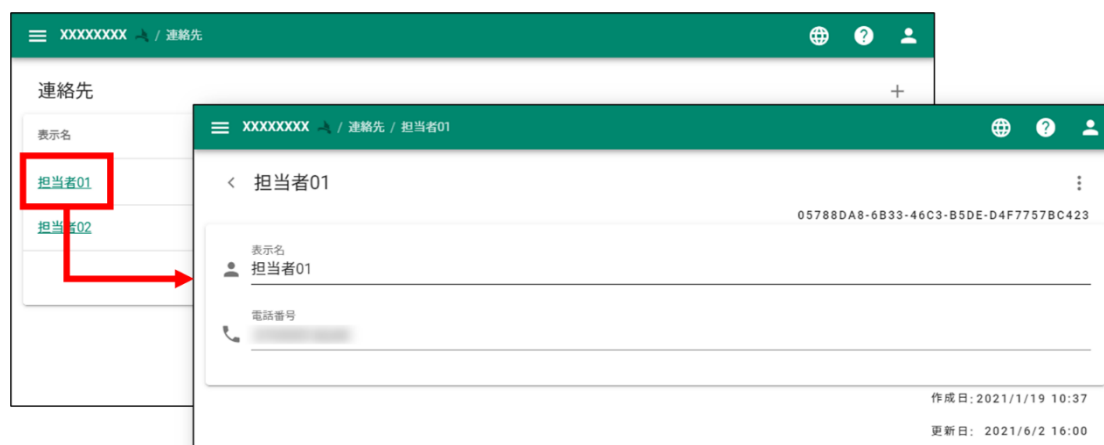
要素	説明
表示名	連絡先の表示名です。
電話番号	連絡先の電話番号です。（1つの連絡先に登録可能な電話番号は1つです。）

4.1.2. 連絡先管理画面

Pigeon のメニューより「連絡先」をクリックすることで、連絡先の管理画面に移動することが可能です。



「表示名」はリンクになっており、クリックすることで連絡先の詳細を確認することが可能です。



4.2. コールフロー

4.2.1. コールフローとは

「コールフロー」とは、電話連絡時の架電順やループ回数について管理するためのオブジェクトです。
以下の要素で構成されています。

要素	説明
表示名	コールフローの表示名です。
架電順	電話連絡を行う順番です。 作成した連絡先の中から、コールフローに加える対象を選択します。
最大ループ回数	電話連絡時にループ可能な回数です。

4.2.2. コールフロー管理画面

Pigeon のメニューより「コールフロー」をクリックすることで、コールフローの管理画面に移動することが可能です。



「表示名」はリンクになっており、クリックすることでコールフローの詳細を確認することが可能です。



4.2.3. 最大ループ回数とは

架電順で最後となる担当者に電話連絡を行ったものの、担当者が電話に出なかった場合等に、再び架電順の先頭から電話連絡をやりなおす機能を「ループ」と呼びます。

「最大ループ回数」とは、ループを何回まで行うかの定義となり、コールフローごとに設定します。

例：架電順が「担当者 01 → 担当者 02」でループ回数「3」の場合、電話連絡は

「担当者 01 → 担当者 02 → 担当者 01 → 担当者 02 → 担当者 01 → 担当者 02」のように行われ、最後の担当者 02 の電話連絡においても担当者が電話に出ない場合等は電話連絡が終了(失敗)します。

4.3. ガイダンス

「ガイダンス」とは、電話連絡時の再生メッセージや応答について管理するためのオブジェクトです。

以下の要素で構成されています。

要素	説明
表示名	ガイダンスの表示名です。
メッセージテンプレート	電話に出た時に再生されるメッセージです。 メッセージ内のパラメータ展開が可能です。（※）
応答	プッシュボタンに対する挙動と再生されるメッセージのセットです。 1つのガイダンス内に複数の応答を定義することが可能です。 (ただし、1回の電話連絡で反応可能な応答は1回までとなります。) 応答のメッセージ内のパラメータ展開が可能です。（※）

※ パラメータ展開については後述します。

4.3.1. ガイダンス管理画面

Pigeon のメニューより「ガイダンス」をクリックすることでガイダンス管理画面に移動することが可能です。

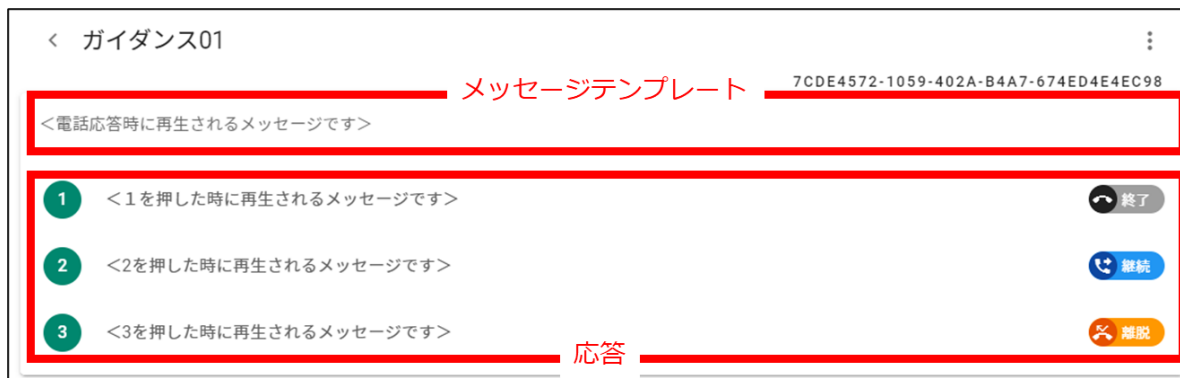


「表示名」はリンクになっており、クリックすることでガイダンスの詳細を確認することが可能です。



4.3.2. 応答とは

「応答」とは、プッシュボタンに対する挙動と、その際に再生するメッセージのセットです。



応答は、1～9 までのボタンに対して以下 3 パターンの「応答タイプ」を定義することが可能です。

応答パターン	挙動
終了	電話連絡を終了します。 電話連絡の結果は「成功」となり、次の担当者には電話連絡が行われません。
継続	次の担当者に電話連絡を行います。 次のループの際、再び自分への電話連絡が行われます。
離脱	次の担当者に電話連絡を行い、自分を離脱させます。 この電話連絡の間、次以降のループの際、自分への電話連絡は行われません。

1つのガイダンスの中で、複数のボタンに対する応答が定義可能となり、複数のボタンに同じ応答タイプを定義することが可能となりますが、最低でも応答タイプが「終了」の応答を1つ定義する必要があります。

4.3.3. ガイダンス内のパラメータ展開

ガイダンス内の「メッセージテンプレート」と、「応答のメッセージ」は、mustache 記法と呼ばれる記法により、電話連絡時のパラメータ展開を行うことが可能です。

各メッセージ内に任意のキー名を用いて「{{キー名}}」と記載したうえ、電話連絡時に、キー名に対する値をパラメータとして渡すことで、電話連絡時に「{{キー名}}」の部分がパラメータ展開され再生されます。

以下は記載方法の例となります。

< ガイダンス

DBF25AC9-8C6D-4486-9935-49D4E846392D

{{server}}で障害が発生しました。障害内容は{{alert}}です。

1

 {{msg}}

終了



電話連絡時に以下パラメータを指定

server : ウェブサーバ
alert : サービス停止
msg : 電話連絡を終了します。

【実際の再生メッセージ】

ウェブサーバで障害が発生しました。障害内容はサービス停止です。

【プッシュボタン1を押した時の実際の再生メッセージ】

電話連絡を終了します。

4.4. 連絡履歴

4.4.1. 連絡履歴確認画面

Pigeon のメニューより「連絡履歴」をクリックすることで、連絡履歴の確認画面に移動することが可能です。

ステータス	架電開始日時	理由	入力キー
成功	2021/6/3 15:40	received response	1
成功	2021/6/3 14:14	received response	1
成功	2021/5/6 17:08	received response	9
成功	2021/5/6 17:08	received response	9
成功	2021/5/6 17:01	received response	9
成功	2021/5/6 17:00	received response	9
成功	2021/5/6 16:57	received response	9

「架電開始日時」はリンクとなっており、クリックすることで連絡履歴の詳細を確認することが可能です。

連絡履歴

ステータス	架電開始日時	理由	入力キー
成功	2021/6/3 15:40	received response	1
成功	2021/6/3 14:14	received response	1
成功	2021/5/6 17:08	received response	9

< 2021/6/3 15:40

5E17FAA3-90C8-4FED-8129-8181E2DAC9C5

1 / 5
ループ回数

1
入力キー

received response
理由

架電履歴

ステータス	架電開始日時	連絡先	電話番号	入力キー	動作
成功	2021/6/3 15:40	担当者01		1	終了

1ページあたりの行数: 10 1-1 件目 / 1件

コールフロー

#1 担当者01

× 5回

ガイドンス

{{(server)}}で障害が発生しました。障害内容は{{(alert)}}です。

1 {{(msg)}} 終了

パラメータ

alert
サービス停止

msg
連絡を終了します

server
ウェブサーバ

連絡履歴の詳細では、以下項目について確認することが可能です。

(画像は、電話連絡が「成功」のステータスで終わった場合の表示例です。)

The screenshot displays a call log for a successful call on 2021/6/3 at 15:40. The interface is divided into several sections:

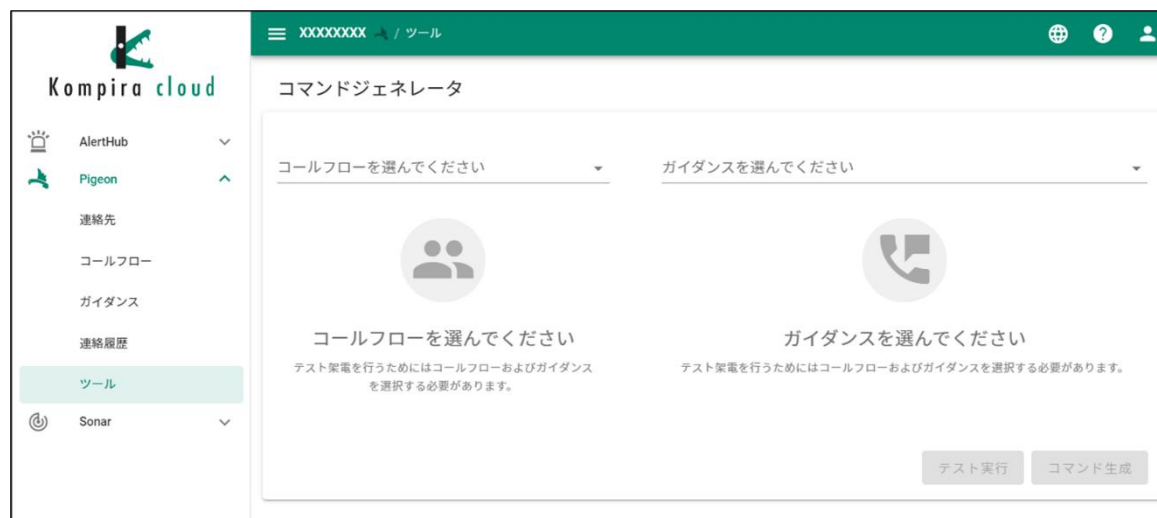
- Header:** Shows a back arrow, the date/time, and a green '成功' (Success) status.
- Summary Bar:** A green bar containing '1 / 5 ループ回数' (1 / 5 Loop count), '1 入力キー' (1 Input key), and 'received response 理由' (received response Reason).
- Call History Table:** A table with columns: ステータス (Status), 架電開始日時 (Call start time), 連絡先 (Contact), 電話番号 (Phone number), 入力キー (Input key), and 動作 (Action). The first row shows a '成功' (Success) status, a start time of 2021/6/3 15:40, contact '担当者01', a redacted phone number, input key '1', and an action of '終了' (End).
- Call Flow Panel (3):** Shows a call flow with step #1 '担当者01' and a total of 5 loops.
- Guidelines Panel (4):** Shows a guideline with a message: '{{server}}で障害が発生しました。障害内容は{{alert}}です。' (A failure occurred on {{server}}. The failure content is {{alert}}.) and a step 1 action: '{{msg}}' (End).
- Parameters Panel (5):** Shows parameters: 'alert' (サービス停止 - Service stop), 'msg' (連絡を終了します - End contact), and 'server' (ウェブサーバ - Web server).

項番	項目	説明
1	結果サマリ	電話連絡の開始時刻や結果とその理由、ループを行った回数や最終的な応答ボタン（押された場合）の確認が可能です。
2	架電履歴	電話連絡時に、実際に電話をかけた全ての担当者が順番に表示され「いつ」「誰に」電話を行い「その結果」の確認が可能です。
3	コールフロー	電話連絡に使用されたコールフローの定義情報の確認が可能です。
4	ガイダンス	電話連絡に使用されたガイダンスの定義情報の確認が可能です。
5	パラメータ	電話連絡時に渡されたパラメータの確認が可能です。

4.5. ツール

4.5.1. ツール利用画面

Pigeon のメニューより「ツール」をクリックすることで、「コマンドジェネレータ」という補助ツールの利用画面に移動することが可能です。



4.5.2. コマンドジェネレータとは

「コマンドジェネレータ」とは、コールフローやガイダンスを選択することで、API 呼び出し用のコマンド生成や、電話連絡テストを実行することが可能となる補助ツールです。

Pigeon は、監視装置やチケット管理システム等、外部システムから呼ばれることが主要な利用方法となります。その為のコマンド生成や、事前の電話連絡テストの実行等に利用することが可能です。



ガイダンスにパラメータが含まれる場合も、コマンドジェネレータによりパラメータの埋め込みを簡単に行うことが可能です。（埋め込んだパラメータは、テスト実行やコマンド生成にも反映されます。）

コールフローを選んでください コールフロー01	ガイダンスを選んでください ガイダンス02
選択されたコールフロー	選択されたガイダンス
#1 担当者01	{{(host)}}にて{{(datetime)}}に{{(alert)}}が発生しました。対応可能場合は1を、対応できない場合は2を、この連絡から離脱する場合は3を押してください。
× 5 回	1 電話連絡を終了します。 2 次の担当者に電話連絡を行います。 3 この電話連絡から離脱します。
	パラメータを入力してください host DBサーバ datetime 2021/01/01 00:00:00 alert サービス停止

5. 設定の流れ

本項では、Pigeon の設定の流れについてチュートリアル形式で説明します。

設定は、以下の順番で実施します。

1. 連絡先の作成
2. コールフローの作成
3. ガイダンスの作成
4. テスト実行
5. テスト結果の確認
6. API トークンの準備
7. API 実行コマンドの生成
8. 外部からの呼び出し

設定例として、今回は以下のような各種データを設定します。

対象	説明
連絡先	「担当者 01」「担当者 02」「担当者 03」の 3 つの連絡先を作成
コールフロー	「担当者 02 → 担当者 01 」の順に、最大で 5 回ループするコールフローを作成
ガイダンス	障害が発生したサーバ名をパラメータとして展開可能であり、プッシュボタン応答が「1：終了」「2：継続」「3：離脱」となるようなガイダンスを作成

5.1. 連絡先の作成

はじめに、「担当者 01」「担当者 02」「担当者 03」の連絡先の作成を行います。

Pigeon のメニューより「連絡先」をクリックして連絡先管理画面へと移動した後、連絡先一覧の右側にある「+」マークをクリックします。



「+」マークをクリックすると連絡先の作成ダイアログが表示されるので、表示名と電話番号を入力し、連絡先を作成します。（以下は、担当者 01 の連絡先を作成している様子です。）

表示名
担当者01 5 / 30

電話番号
[masked] 11 / 30

保存

連絡先を作成すると、以下のように作成した連絡先の詳細画面へと移動します。
画面左上「<」マークをクリックすることで、連絡先の管理画面へと戻ることが可能です。

Kompira cloud

AlertHub
Pigeon
連絡先
コールフロー
ガイダンス
連絡履歴

担当者01

CACF6B1E-5C49-4762-8B67-2FC938EFC9AC

表示名
担当者01

電話番号
[masked]

作成日: 2021/6/4 14:35
更新日: 2021/6/4 14:35

作成した連絡先は、連絡先管理画面の一覧に表示されるようになります。

連絡先

表示名	電話番号	
担当者01	[masked]	

1ページあたりの行数: 10 1-1 件目 / 1件

同じように、担当者 02 と担当者 03 についても連絡先の作成を行きましょう。

作成が終了すると、連絡先管理画面には 3 人の担当者の連絡先が一覧表示されるようになります。

表示名	電話番号	
担当者03		
担当者02		
担当者01		

初期状態では、作成されたデータは「後に作成されたもの」が上段に表示されます。

一覧の各タイトルをクリックすることで、そのタイトルをキーとした並び替えを行うことが可能です。



5.2. コールフローの作成

次に、コールフローの作成を行います。

Pigeon のメニューより「コールフロー」をクリックしてコールフロー管理画面へと移動した後、コールフロー一覧の右側にある「+」マークをクリックします。



「+」マークをクリックするとコールフローの作成ダイアログが表示されるので、まずは表示名とループ回数を入力します。

The screenshot shows the Call Flow creation dialog box. The '表示名' (Display Name) field is filled with 'コールフロー' and is highlighted with a red box. The '最大ループ回数' (Maximum Loop Count) field is filled with '5'. The '連絡先' (Contact) field is empty. At the bottom, there are 'キャンセル' (Cancel) and '保存' (Save) buttons.

続いて架電順の登録です。

連絡先のリストボックスのプルダウンをクリックすると、作成した連絡先がリスト表示されるので、「担当者02」をクリックし、1 番目の電話連絡先として登録します。

表示名
コールフロー 6 / 30

最大ループ回数
5

連絡先

担当者03
担当者02
担当者01

キャンセル 保存

再び連絡先のリストのプルダウンをクリックしたうえ、今度は「担当者01」をクリックし、2 番目の連絡先に追加されたことを確認したのち「保存」ボタンをクリックしコールフローを作成します。

表示名
コールフロー 6 / 30

最大ループ回数
5

連絡先
担当者02
08043275784

担当者03
担当者02
担当者01

担当者02
担当者01

キャンセル 保存

コールフローを作成すると、以下のようにコールフローの詳細画面へと移動します。

画面左上「<」マークをクリックすることで、コールフローの管理画面へと戻ることが可能です。

Kompira cloud

AlertHub
Pigeon
Sonar

コールフロー

#1 担当者02
#2 担当者01
× 5 回

作成日: 2021/6/4 17:46
更新日: 2021/6/4 17:46

作成したコールフローは、コールフロー管理画面の一覧に表示されるようになります。



表示名	架電順	最大ループ回数
コールフロー	担当者02 → 担当者01	5 回

今回は、連絡先で作成した担当者 03 は、コールフロー作成時に架電順に含まれていません。
この場合、作成したコールフローを利用した電話連絡において、担当者 03 には電話連絡は行われません。

このように、連絡先は、コールフローの中の架電順に登録することで、はじめて電話連絡時の連絡対象として扱われます。

5.3. ガイダンスの作成

最後に、ガイダンスの作成を行います。

今回は、障害が発生したサーバ名をパラメータで展開可能なメッセージテンプレートで、1～3 までのプッシュボタンに対し、それぞれ「終了」「継続」「離脱」となるようガイダンスを作成します。

Pigeon のメニューより「ガイダンス」をクリックしてガイダンス管理画面へと移動した後、ガイダンス一覧の右側にある「+」マークをクリックします。



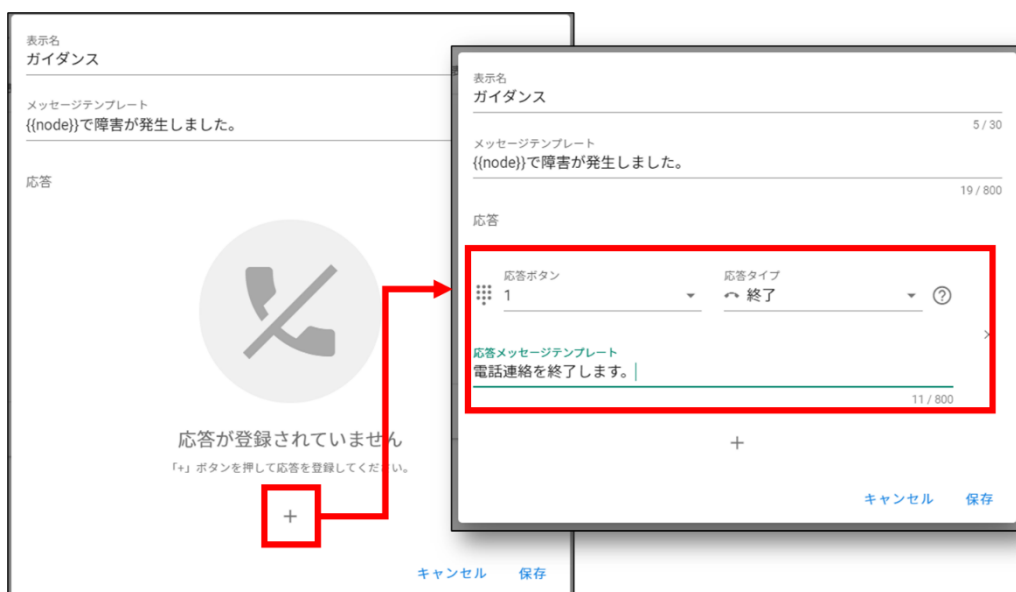
「+」マークをクリックするとガイダンスの作成ダイアログが表示されるので、まずは表示名とメッセージテンプレートを入力します。



次に、応答の作成を行います

ダイアログ下部の「+」をクリックすると、応答を追加することができるようになるので、応答ボタンと応答タイプを選択したうえ、応答メッセージテンプレートを入力します。

(以下画像は、1 が押された時に電話連絡を終了する応答を作成している様子です。)



同じように、2 が押された時に電話連絡を継続（次の担当者に電話連絡）する応答を作成します。

表示名
ガイダンス

メッセージテンプレート
{{node}}で障害が発生しました。

応答

応答ボタン 1 応答タイプ 終了

応答メッセージテンプレート
電話連絡を終了します。

+

表示名
ガイダンス

メッセージテンプレート
{{node}}で障害が発生しました。

応答

応答ボタン 1 応答タイプ 終了

応答メッセージテンプレート
電話連絡を終了します。

+

応答ボタン 2 応答タイプ 継続

応答メッセージテンプレート
次の担当者に連絡します。

+

最後に、3 が押された時に電話連絡を離脱（次の担当者に電話連絡+自分は離脱）する応答を作成したうえ保存ボタンをクリックしガイダンスを作成します。

メッセージテンプレート
{{node}}で障害が発生しました。

応答

応答ボタン 1 応答タイプ 終了

応答メッセージテンプレート
電話連絡を終了します。

+

応答ボタン 2 応答タイプ 継続

応答メッセージテンプレート
次の担当者に連絡します。

+

応答ボタン 3 応答タイプ 離脱

応答メッセージテンプレート
次の担当者に連絡し、この電話連絡から離脱します。|

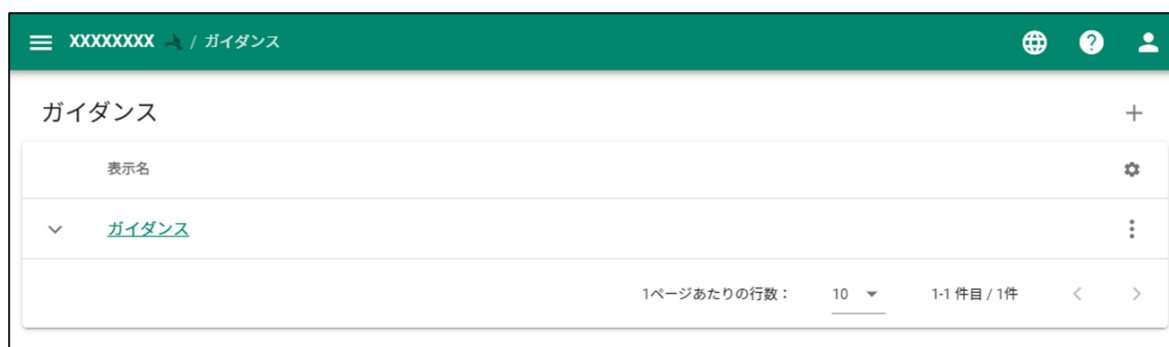
保存

ガイドンスを作成すると、以下のように作成したガイドンスの詳細画面へと移動します。

画面左上「<」マークをクリックすることで、ガイドンスの管理画面へと戻ることが可能です。



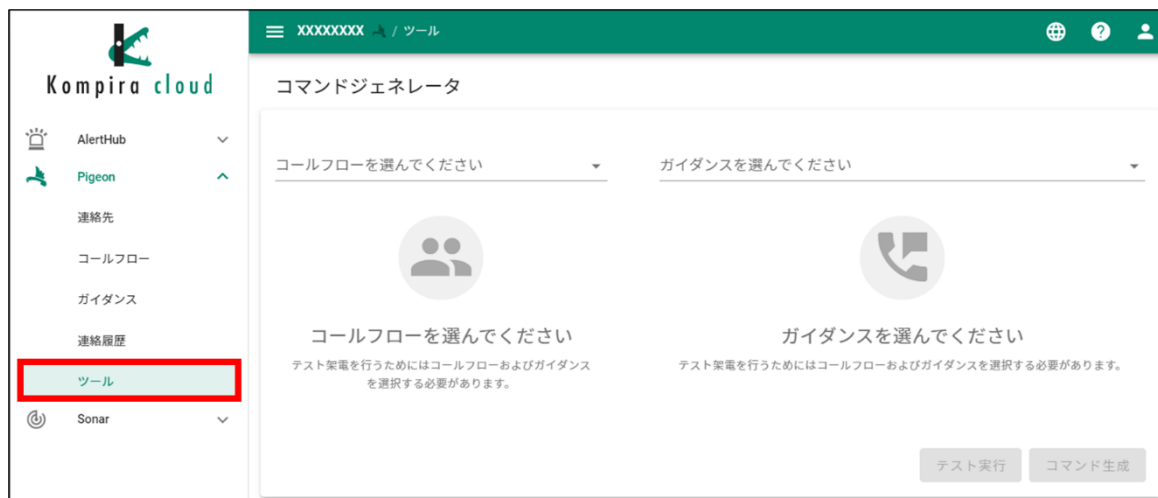
作成したガイドンスは、ガイドンス管理画面の一覧に表示されるようになります。



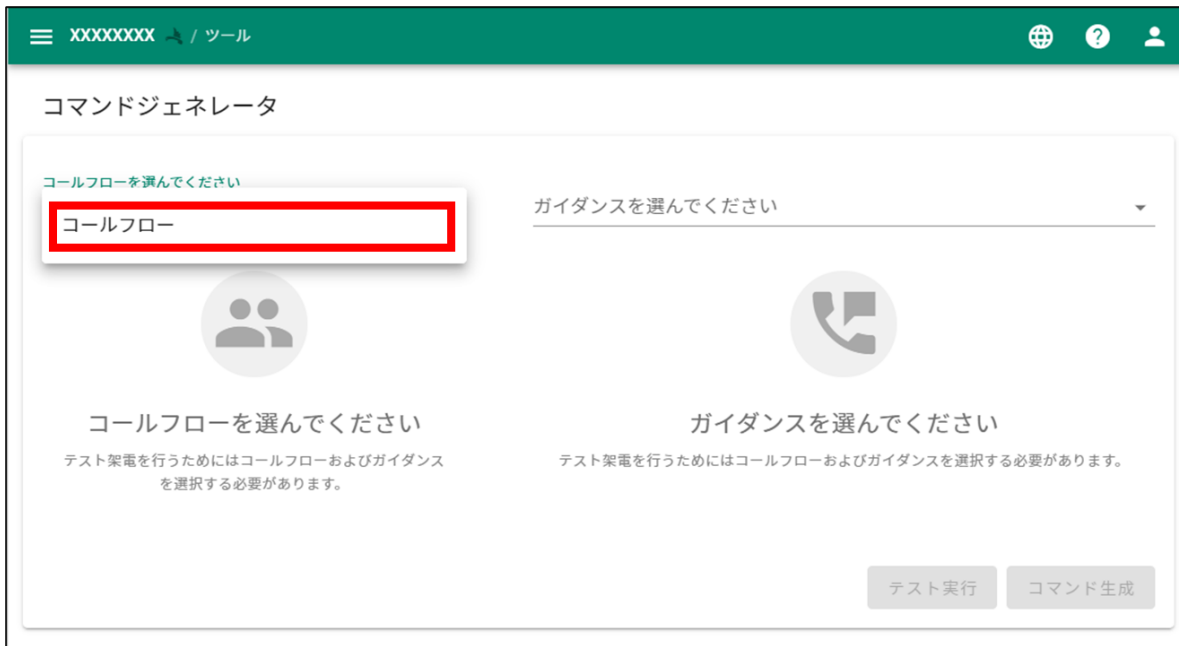
5.4. テスト実行

電話連絡を行う為に必要なオブジェクトの設定が終了したので、一度テストを実行してみましょう。

Pigeon のメニューより「ツール」をクリックしてコマンドジェネレータの画面へと移動します。



まずはコールフローのリストボックスをクリックし、作成した「コールフロー」を選択します。



コマンドジェネレータ

コールフローを選んでください

ガイダンスを選んでください

コールフロー

ガイダンスを選んでください

テスト実行

コマンド生成

コールフローを選択すると、コールフローの詳細が下部に表示されます。



コマンドジェネレータ

コールフローを選んでください

ガイダンスを選んでください

選択されたコールフロー

#1 担当者02

#2 担当者01

x 5回

ガイダンスを選んでください

テスト実行

コマンド生成

次に、ガイダンスのリストボックスをクリックし、前項までに作成した「ガイダンス」を選択します。



コールフロー同様に、下部にガイダンスの詳細が表示されます。

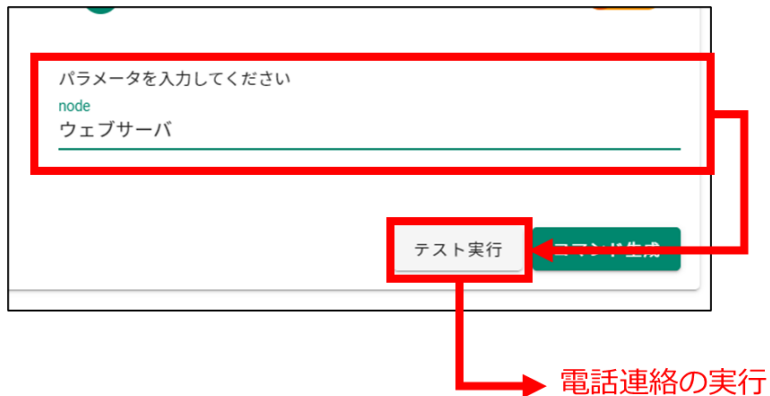
なお、今回作成したガイダンスは、パラメータ「node」をガイダンスのメッセージテンプレートに含むため、併せてパラメータの入力ボックスが表示されます。

(パラメータを含まないガイダンスを選択した場合は、パラメータの入力ボックスは表示されません。)



それでは、実際に電話連絡を実行します。

パラメータ `node` の入力ボックスに、任意の値（画像では「ウェブサーバ」）を入力したうえ、「テスト実行」ボタンをクリックすると、電話連絡のテスト実行が行われます。



テスト実行を開始すると、画面は連絡履歴の詳細画面へと移動します。

（以下は、電話連絡が開始される前の準備中の状態です。）



次項にて、このテスト連絡結果の確認を行います。

5.5. テスト結果の確認

コマンドジェネレータで実行したテスト結果を確認します。

前項にてテスト実行を行った際に表示された画面（連絡履歴の詳細画面）を利用します。

もし画面を移動してしまった場合、Pigeon のメニューより「連絡履歴」をクリックして連絡履歴の確認画面へと移動した後、テスト実行を行った日時と同じ時刻の「架電開始日時」をクリックすることでも連絡履歴の詳細画面を表示することが可能です。

The screenshot shows the Kompira cloud interface. On the left, the '連絡履歴' (Call History) menu item is highlighted with a red box. A red arrow points from this menu item to the '連絡履歴' page. The '連絡履歴' page shows a list of call log entries. The first entry is highlighted with a red box, showing the date and time '2021/6/7 13:03'. A red arrow points from this entry to the detailed view of the call log. The detailed view shows the call log entry for '2021/6/7 13:03' with a status of '成功' (Success). The entry is part of a loop of 3 out of 5 iterations. The call log entry shows the status '成功' (Success), the start time '2021/6/7 13:04', the caller '担当者02', the phone number, the input key '1', and the action '終了' (End). Below the call log entry, there is a table with columns: ステータス (Status), 架電開始日時 (Call Start Time), 連絡先 (Contact), 電話番号 (Phone Number), 入力キー (Input Key), and 動作 (Action). The table contains four rows of call log entries, all with a status of '成功' (Success). The first row shows '担当者02' as the caller, '2021/6/7 13:04' as the start time, and '終了' (End) as the action. The second row shows '担当者02' as the caller, '2021/6/7 13:04' as the start time, and '継続' (Continue) as the action. The third row shows '担当者01' as the caller, '2021/6/7 13:03' as the start time, and '断線' (Disconnection) as the action. The fourth row shows '担当者02' as the caller, '2021/6/7 13:03' as the start time, and '継続' (Continue) as the action.

詳細画面上部では、電話連絡が3回目のループにて1番のボタンを押されて終了したことが確認できます。

The screenshot shows the detailed view of a call log entry for '2021/6/7 13:03'. The entry is part of a loop of 3 out of 5 iterations. The call log entry shows the status '成功' (Success), the start time '2021/6/7 13:04', the caller '担当者02', the phone number, the input key '1', and the action '終了' (End). Below the call log entry, there is a table with columns: ステータス (Status), 架電開始日時 (Call Start Time), 連絡先 (Contact), 電話番号 (Phone Number), 入力キー (Input Key), and 動作 (Action). The table contains four rows of call log entries, all with a status of '成功' (Success). The first row shows '担当者02' as the caller, '2021/6/7 13:04' as the start time, and '終了' (End) as the action. The second row shows '担当者02' as the caller, '2021/6/7 13:04' as the start time, and '継続' (Continue) as the action. The third row shows '担当者01' as the caller, '2021/6/7 13:03' as the start time, and '断線' (Disconnection) as the action. The fourth row shows '担当者02' as the caller, '2021/6/7 13:03' as the start time, and '継続' (Continue) as the action.

架電履歴を見る事で、以下のような流れで電話連絡が行われ、終了したことが確認できます。

(デフォルトの状態では、架電履歴は最新のものが上部に表示されます)

- ・ 担当者 02 (ループ 1 回目) : 2 ボタンを押して継続を選択 (次の担当者に連絡)
↓
- ・ 担当者 01 (ループ 1 回目) : 3 ボタンを押して離脱を選択 (次の担当者に連絡して自分は離脱)
↓
- ・ 担当者 02 (ループ 2 回目) : 2 ボタンを押して継続を選択 (次の担当者に連絡)
↓
- ・ 担当者 02 (ループ 3 回目) : 1 ボタンを押して終了を選択 (電話連絡が終了)

架電履歴						
ステータス	架電開始日時	連絡先	電話番号	入力キー	動作	
成功	2021/6/7 13:04	担当者02		1	終了	
成功	2021/6/7 13:04	担当者02		2	継続	
成功	2021/6/7 13:03	担当者01		3	離脱	
成功	2021/6/7 13:03	担当者02		2	継続	

1ページあたりの行数: 10 1-4 件目 / 4件

担当者 01 は、最初の電話連絡時に離脱を選択したため、次ループ以降は電話連絡がスキップされています。

また、連絡履歴の詳細画面では、コールフローやガイダンスの詳細、電話連絡時に指定されたパラメータも含めて確認を行うことが可能です。

コールフロー	ガイダンス	パラメータ
#1 担当者02 #2 担当者01 × 5 回	{{(node)}}で障害が発生しました。 1 電話連絡を終了します。 終了 2 次の担当者に連絡します。 継続 3 次の担当者に連絡し、この電話連絡から離脱します。 離脱	node ウェブサーバ

これらの情報や、実際の読み上げ内容に基づき、Pigeon の設定が意図されたとおりになっていることが確認できたら、外部からの電話連絡を行うため API トークンの発行やコマンド生成に進みます。

5.6. API トークンの準備

外部からの電話連絡を行えるようにするためには、API トークンを発行する必要があります。

すでに API トークンを発行済みで、利用可能な API トークンがあればそちらを利用し、もし新規に発行する必要がある場合は、共通メニュー利用ガイドを参照し、新規に API トークンの払い出しを行いましょう。

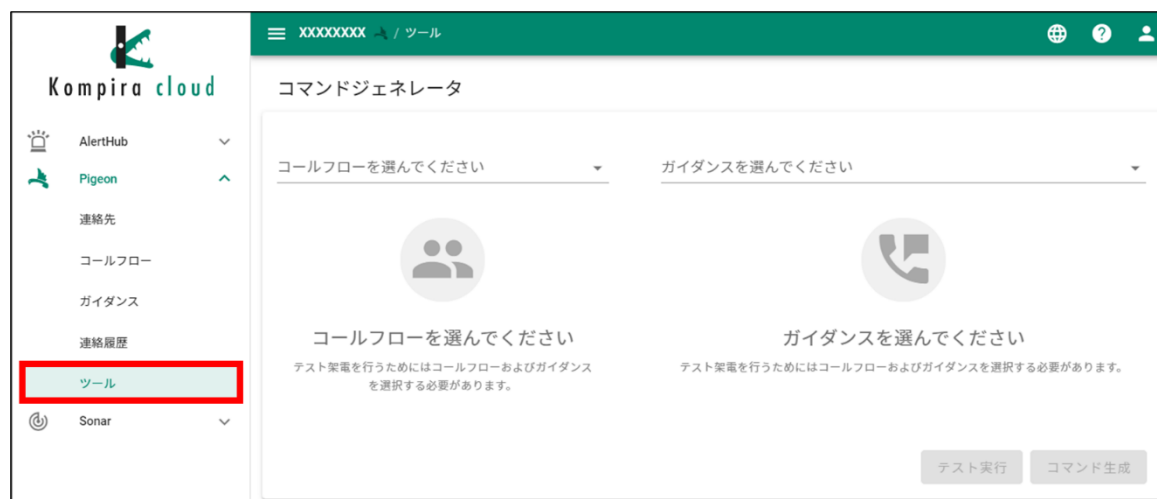
以下は簡易的な API トークンの発行手順です。

簡易手順

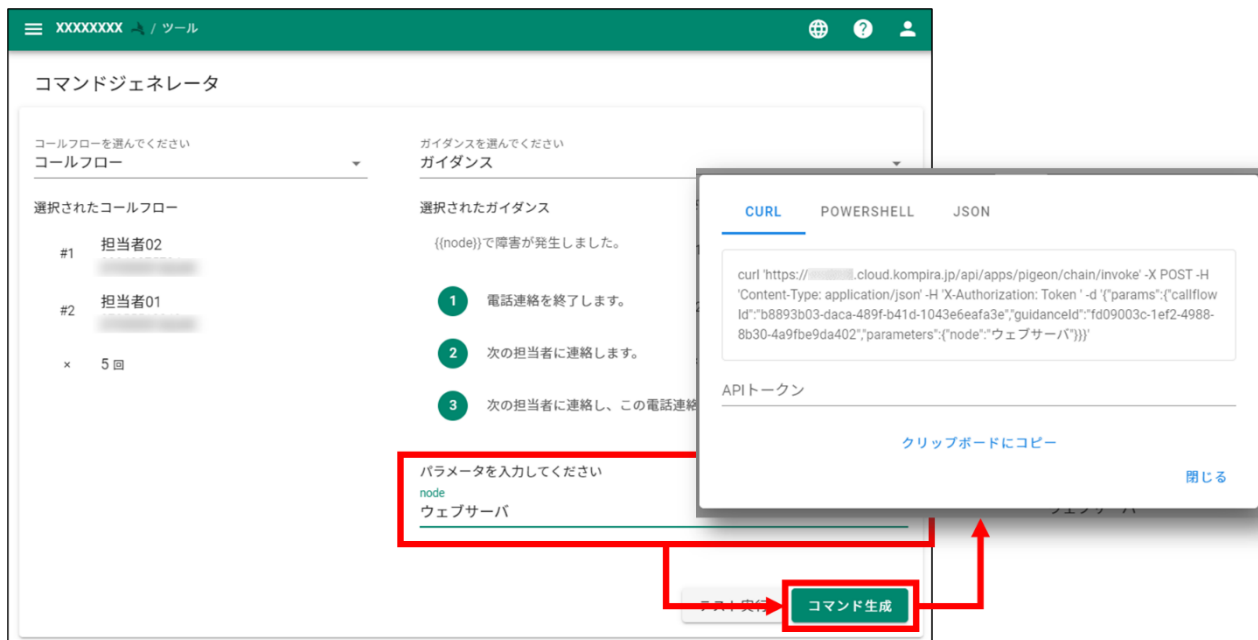
- WEBGUI 右上の人型アイコンクリック → 「設定」にて設定ページに移動
- 「API トークン」をクリックし、表示された API トークン一覧の右側にある「+」をクリック
- ダイアログにラベルと有効期限を入力後、「保存」ボタンをクリックして API トークンを発行

5.7. API 実行コマンドの生成

API 実行コマンドの生成を行うため、再び Pigeon のメニューより「ツール」をクリックしてコマンドジェネレータの画面へと移動します。



テスト実行時と同じように、コールフローとガイダンスを選択し、パラメータに任意の値を入力したうえ、「コマンド生成」ボタンをクリックすると、コマンドが記載されたダイアログが表示されます。



ダイアログ上部の言語をクリックすると、CURL/POWERSHELL/JSON の3つのコマンドを切り替えることが可能です。

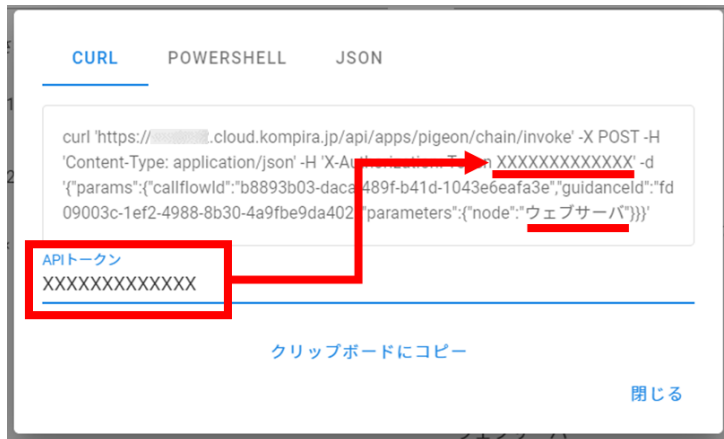
CURLとPOWERSHELLは、APIトークンを入力することでコピー＆ペーストでそのまま使用することが可能なコマンドで、JSONは、外部プログラムからAPIにて電話連絡を行う際、コールフローやガイダンス、パラメータを指定するためのJSON文字列となります。



今回は、例として CURL コマンドを利用します。

予め準備した API トークンをテキストボックスに入力すると、コマンドの中にトークンが反映されます。

また、コマンドジェネレータで指定したパラメータ「ウェブサーバ」についても、コマンドの中にパラメータとして埋め込まれていることが確認できます。



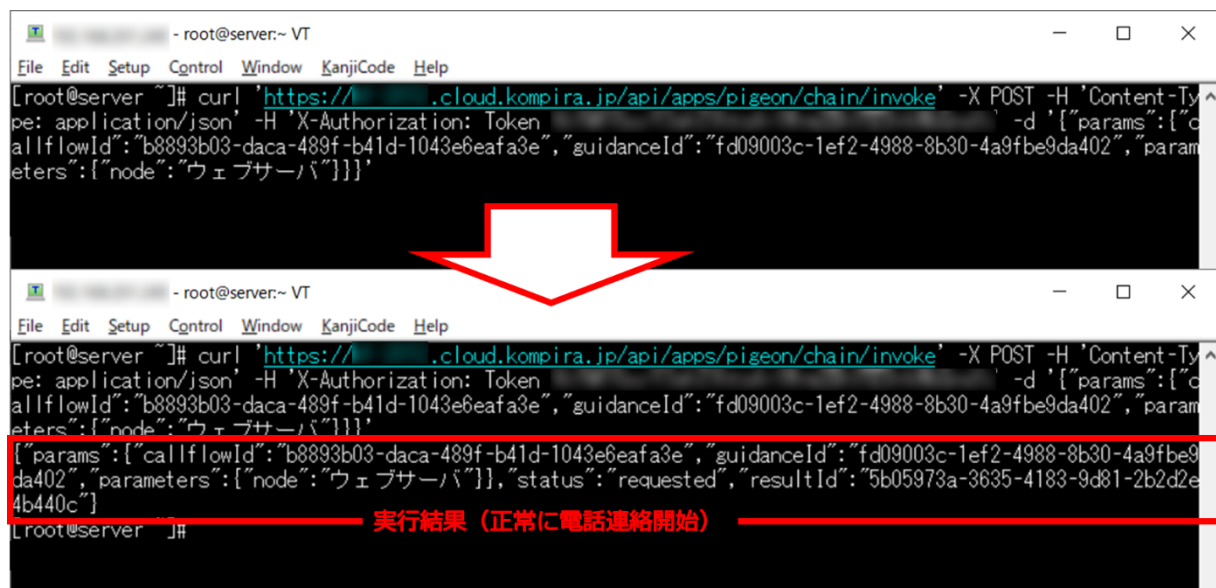
このコマンドをコピーすることで、curl を使用可能な、インターネットに接続されているシステムやサーバから、コマンドジェネレータで指定した内容で電話連絡を行うことが可能となります。

5.8. 外部からの呼び出し

最後に、前項で生成したコマンドを利用し、外部から curl コマンドにて電話連絡を実行してみましょう。

以下は、curl コマンドが実行可能なサーバに SSH 接続し、コピーしたコマンドをペーストして実行した例です。コマンドジェネレータからのテスト実行時と同じように電話連絡が開始されます。

(「ウェブサーバ」を別の文字列に変更すると、読み上げ時に node に展開される文字列も変化します。)



以上が設定の流れとなります。

6. ガイダンス事例紹介

本項では、いくつかの事例に基づいたガイダンスの設定例について紹介します。

- ・ 事例①…使いまわしが容易なガイダンスを作成する
- ・ 事例②…電話終了のパターンを複数用意する

6.1. 事例 ①…使いまわしが容易なガイダンスを作成する

これまでに記載したとおり、ガイダンスや応答のメッセージテンプレートでは、パラメータの展開を行うことが可能ですが、発生したアラートによっては、メッセージテンプレートにおいて「どこの部分に何のパラメータを配置/展開するか」が決定しにくいことがあります。

以下は、汎用的に「毎回メッセージテンプレートを丸ごと生成する」ことを意識したガイダンスの設定例となります。



ガイダンスのメッセージテンプレートをパラメータ（上記例では`{{msg}}`）だけにします。

こうすることで、毎回電話連絡を実行する際に、読み上げ内容を自由に変更することが可能となります。

6.2. 事例 ②…電話終了のパターンを複数準備する

運用自動化を構築する際、「アラートが発生し、電話にてユーザにその旨報告するとともに、その後のオペレーションを選択させたい」といった要望が発生することがあります。

以下は、そういった要望を満たすためのガイダンスの設定例となります。



応答パターンは複数作成することが可能な為、異なる「終了タイプ」を持つ応答を作成します。

こうすることによって、同じ「終了」でも「何番を押して終了したか」により、その後のオペレーションを分岐させる判断要素にすることが可能となります。

なお、電話結果の取得を WEBGUI 以外から行う為には、API を使用する必要があります。

また、「電話連絡サービス」である Pigeon では、障害対応等の運用オペレーションそのものの自動化は行えず、連携システムの構築が必要となるため、本事例は高度な運用自動化を見据えた設定事例となります。

7. 補足資料

7.1. API について

API を利用することで、電話連絡だけでなく、連絡先やコールフロー、ガイドンスの作成や、電話連絡履歴の取得など、WEBGUI から行える操作の全てを行うことが可能となります。

また、Pigeon では、以下のような操作は API からでのみ実施可能となります。

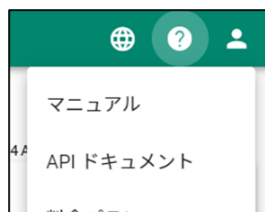
- ・ 電話連絡の際、一時的な連絡先やコールフロー、ガイドンスを作成して電話連絡を行う
- ・ 電話連絡が終了した際、終了時のステータス（成功/失敗）等を含めた通知を行う

なお、「設定の流れ」で記載したとおり、API の実行には API トークンの発行が必要となります。

7.2. API ドキュメント

API の一覧や動作概要等のドキュメントは、WEBGUI の関連リンクより参照することが可能です。

関連リンクは、WEBGUI 右上「」より確認可能です。



7.3. API 実行とコマンド構文例

API の実行は、行う操作により URL やメソッド、パラメータの有無や種類が異なりますが、共通要素として、ヘッダ情報に以下を含めることが必要になります。

ヘッダ項目	内容
Content-Type	application/json
X-Authorization	Token API トークン (「Token XXXXXXXXXXXXXXXXXXXX」のように記載)

また、実行する API ごとに、URL やメソッド、パラメータとして指定する項目が異なります。
これらは、API ドキュメントにて詳細を確認することが可能です。

以下は、「設定の流れ」にて、コマンドジェネレータで生成された、電話連絡 API を curl にて実行するコマンド例です。(スペース ID や API トークンは一部伏字にしています。)

```
curl 'https://xxxxxxx.cloud.kompira.jp/api/apps/pigeon/chain/invoke' -X POST  ¥  
  -H 'Content-Type: application/json' -H 'X-Authorization: Token xxxxxxxxxxxxxxxxxxxx'  ¥  
  -d '{"params":{"callflowId":"b8893b03-daca-489f-b41d-1043e6eafa3e","guidanceId":"fd09003c-1ef2-4988-8b30-4a9fbe9da402","parameters":{"node":"ウェブサーバ"}}}'
```

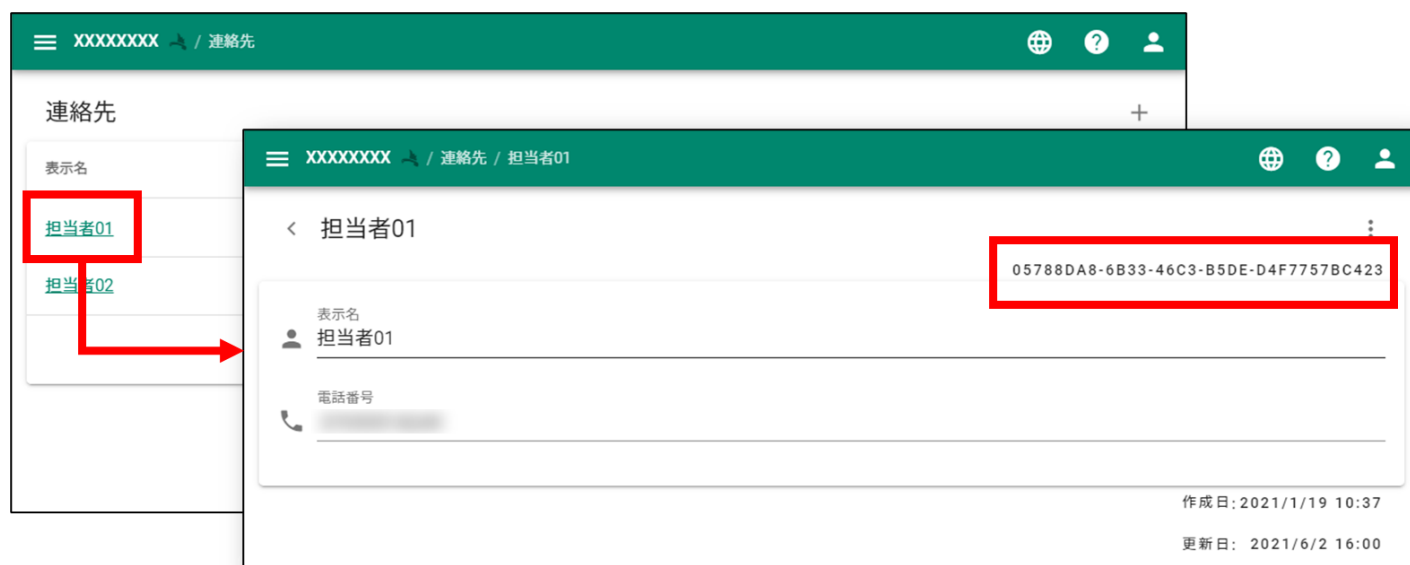
電話連絡 API は、URL 「https://<ベース URL>/api/apps/pigeon/chain/invoke」 に対してメソッド POST にてアクセスを行い(-X 部分)、パラメータとして「コールフローID」「ガイダンス ID」「展開用パラメータ」が指定(-d 部分)され、実行されることになります。

7.4. オブジェクト ID について

前項に「コールフローID」や「ガイドンス ID」等の単語が出てきましたが、これらは「Kompira cloud 内で、オブジェクトを一意に識別する ID」となり、連絡先や連絡履歴含め、様々なオブジェクトに付与され、特に API 利用時に意識/利用されるものとなります。

各 ID は、オブジェクトの詳細画面内、オブジェクト表示名の右側に UUID 形式で表示されています。

以下画像は、連絡先における連絡先 ID の表示位置の例となりますが、コールフローやガイドンス、連絡履歴についても同様の方法で ID の確認が可能です。



ID 部分をクリックすると、ID をクリップボードにコピーすることが可能です。

