

Confero API/CoCon API 補足資料

目次：

- [1. Confero API について](#) (Plixus 7.x/Confidea G4)
- [2. CoCon API について](#) (Pluxus 6.x/CoCon Server)

1. Confero API (Plixus 7.x/Confidea G4)

Confero API はこちら <https://tcs-static.azurewebsites.net/confero-customer-api/index.html>
バージョン互換： Plixus AE-R/MME バージョン 7.x 以上 / ConfideaWAP G4 (スタンドアローン動作)
*Plixus CRP 6.x (CoCon バージョン) 以下についてはプロトコルが異なります。「CoCon API について」を参照してください

【コマンド形式】

- ✓ Confero API による制御コマンドの伝送には HTTP/HTTPS を用います。Plixus エンジン/Confidea WAP G4 が サーバー、制御器がクライアントとなります。
- ✓ Confero API は REST/JSON フォーマット、Long polling 方式を採用しております。
- ✓ Confero 360 API は HTTPS 暗号化通信を実装しています。HTTPS の利用にはサーバーに証明書をアップロードする必要があります (WEB GUI 操作による) 証明書の発行については Swagger UI に記述された説明を参照してください。
 - サーバーに証明書がアップロードされていない場合、通信は HTTP で、ポート番号は 9080 です。
 - サーバーに証明書がアップロードされている 場合、通信は HTTPS で、ポート番号は 9443 です。

【Swagger UI】

- ✓ Confero API 仕様は「Swagger UI」にて記述されています (英文)
Google Chrome 等の Web ブラウザに以下のアドレスを入力して Swagger UI にアクセスしてください。
`http://{サーバーIP}/openapi/`
- ✓ Swagger UI の API ターミナル機能を使用して制御の確認が可能です。
Swagger UI の「Authorize」ボタンをクリックするとトークン入力画面が現れるので、トークンを入力して Authorize します。
これにより Swagger UI から生成される各種 HTTP コマンドのヘッダーにトークンがセットされます。

【API アクセスキー/トークン】

- ✓ Confero API アクセスには HTTP ヘッダーにセキュリティトークン (アクセスキー) をセットする必要があります。 (トークン様式一例： 08b64f33-01f2-400a-865b-af552da01c94)
- ✓ トークンは Plixus エンジン/WAP G4 の API 設定 (WEB GUI による) にて発行します。

【 Notification について 】

- ✓ televic デバイスのイベントを制御クライアントへステータスポーリングするコマンドです。ステータスポーリングをひとつ取得するために、「Notification」 (通知をもらうコマンド) を送る必要があります。
- ✓ Plixus/CoCon のイベント制御、もしくはステータスの GET コマンドは Notification を必要としません。Notification は include-filter オプションにより必要な Event のみ取得することが可能です。Event は module によって、任意に選択可能です。
 - Audio: システムの音声設定
 - Discussion: 発言ステータスの変更/取得、会議設定変更などのオペレーター機能
 - Room: 会議室設定にアクセスするためのモジュール
 - Meeting: 会議制御のためのモジュール

(以上)

2. Cocon API (Plixus 6.x/CoCon Server)

Cocon API はこちら <https://documents.televic.digital/conference/index.php/s/2pr5B42b6SAw6d3>

- ✓ バージョン互換：Plixus CRP 6.x / CoCon6.x
- ✓ Confero (Plixus CRP 7.x) についてはプロトコル仕様が異なります。「Confero API について」を参照してください。

REST/JSON

Plixus の HTTP API は REST/JSON フォーマット、Long polling 方式を採用しております。

- ✓ ”Server to client” が Connect しておくでプッシュされるコマンドです。
- ✓ ”Client to server” が制御器から送る問い合わせです。

API module compatibility

API 資料の「API module compatibility」リストのチェックボックスを確認してください。

- ✓ 「Plixus Core」にチェックのある項目は Plixus エンジンにターゲットに制御可能な項目です。
- ✓ 「CoCon for Plixus」にチェックのある項目は CoCon Server (サーバーPC) をターゲットに制御可能な項目です。

例：SetState (マイクオンオフ) は「Plixus Core」に該当し Plixus エンジンに直接制御可能です

Connect/Notification

「Server to Client」に該当するコマンドは Plixus/CoCon のイベントを制御クライアントへステータスポーリングするコマンドです。

- ✓ ステータスポーリングを取得するために、はじめ「Connect」(ロングポーリングのスタート)を送る必要があります。
- ✓ ステータスポーリングをひとつ取得するために、「Notification」(通知をもらうコマンド)を送る必要があります。
- ✓ 「Server to Client」に該当するコマンドは Plixus/CoCon のイベント制御もしくはステータスの GET コマンドで、「Connect」「Notification」を必要としません。
- ✓ 「Server to Client」「Client to Server」のセッションは、異なるポートを使用した独立したものである必要があります。(送信元ポート1つだと、ステータスポーリングを待つセッションにイベント制御コマンドが送信され、そのコマンドが処理されません。

CoCon API Test Tool

CoCon API Test Tool は Plixus/CoConServer 用 制御テスト/ターミナルツールです。制御検証に活用ください。

CoCon API Test Tool は、CoCon ソフトウェアセット (CoCon Server/CoCon クライアントアプリケーション) に組み込まれております。下記 URL から最新版のダウンロードが可能です。

<https://www.televic.com/en/conference/support/software-updates>

- ✓ Plixus 専用ソフトウェア CoCon をインストールし、[インストール先ディレクトリ]¥CoCon API Test Tool¥CoCon API Test Tool.exe を起動すると CoCon API Test Tool のウィンドウが現れます。
- ✓ Communication Log には、テストツール⇄主装置 間のコマンド送受が履歴されます。
- ✓ 「Client to Server」コマンドのテスト： URI Command に「~CoCon/」以下のコマンドを入力し Request をクリックするとコマンドが送信されます。例えば SetState を使って Seat1 をオンする場合、「Microphone/SetState/?State=On&SeatNr=1」を入力し Request をクリックします。
- ✓ 「Server to Client」コマンドのテスト (Connect/Notification セッションのテスト)： Plixus エンジンに制御のターゲットとする場合、Base URL に [http://\[Plixus エンジンの IP アドレス\]:8890/CoCon/](http://[Plixus エンジンの IP アドレス]:8890/CoCon/) を入力します。CoCon Server (PC) を制御のターゲットとする場合、Base URL に [http://\[CoCon サーバー \(PC\) の IP アドレス\]:8890/CoCon/](http://[CoCon サーバー (PC) の IP アドレス]:8890/CoCon/) を入力します。Connect をクリックすると Connect コマンドが制御のターゲットに送信されます。Connect が成功し API 制御が成立するとステータス取得が可能になります。ステータスをひとつ取得するたびに、Test Tool から自動的に Notification がひとつ送信されます。

(以上)