Plat' Home
TECHNOLOGY to serve you.

AirManage 2

サブスクリプションプラン向け

ユーザーズガイド

バージョン 1.04
ぷらっとホーム株式会社

■ 商標について

- 文中の社名、商品名等は各社の商標または登録商標である場合があります。
- その他記載されている製品名などの固有名詞は、各社の商標または登録商標です。

■ 使用にあたって

- 本書の内容の一部または全部を、無断で転載することはご遠慮ください。
- 本書の内容は予告なしに変更することがあります。
- 本書の内容については正確を期するように努めていますが、記載の誤りなどにご指摘がございましたら弊社サポート窓口へご連絡ください。
  また、弊社公開の WEB サイトにより本書の最新版をダウンロードすることが可能です。
- 本装置の使用にあたっては、生命に関わる危険性のある分野での利用を前提とされていないことを予めご了承ください。
- その他、本装置の運用結果における損害や逸失利益の請求につきましては、上記にかかわらずいかなる責任も負いかねますので予めご了承ください。

目次

1. はじめに

1.1. 特徴

AirManage 2 は、OpenBlocks IoT Family や EasyBlocks の一括管理を実現するためのシステムです。OpenBlocks IoT Family や EasyBlocks の運用には、個々のユニットの WEB UI での設定やその変更が必要です。したがって、運用台数が増えるにつれ管理が煩雑となります。AirManage 2 を導入すると、AirManage 2 サーバの WEB UI から一括管理することができます。

1.1.1. リコンフィグ

AirManage 2 を導入すると、OpenBlocks IoT Family や EasyBlocks の WEB UI で個々に設定する必要がなくなります。OpenBlocks IoT Family や EasyBlocks のユニットを設置し、AirManage 2 サーバに接続すると、AirManage 2 サーバから設定ファイルなど必要なファイルを取得して、自動的に AirManage 2 の管理下に入ります。この機能をリコンフィグと呼びます。リコンフィグにより AirManage 2 の管理下に入った OpenBlocks IoT Family や EasyBlocks のユニットをノードと呼びます。

1.1.2. パッケージのバージョン管理

AirManage 2 では、ノードのパッケージのバージョンを一括管理できます。パッケージのダウンロードと更新を計画的に実行することができます。

1.1.3. サポートログの取得

OpenBlocks IoT Family や EasyBlocks のサポートを受けるときに必要なサポートログは、AirManage 2 サーバの WEB UI から取得できます。

1.1.4. 常時接続とオンデマンド接続

AirManage 2 サーバとノードの接続形態には、常時接続とオンデマンド接続の二種類があります。

1.1.4.1. 常時接続

ノードが AirManage 2 サーバに常に接続している形態です。

1.1.4.2. オンデマンド接続

指定した時間内に指定した間隔で、ノードが AirManage 2 サーバに接続する形態です。通信料金を抑えたい場合などに使用します。

1.1.5. Web アクセス

AirManage 2 サーバからノードの WEB UI にアクセスできます。

1.2. 管理概要

AirManage 2 は、AirManage 2 サーバとノードからなります。各ノードは、テナントとその下層にあるグループの二階層からなる所属先に配置されます。

<table>
<tr><td colspan="7">AirManage 2 サーバ</td></tr>
<tr><td colspan="3">テナント</td><td colspan="4">テナント</td></tr>
<tr><td colspan="3">グループ</td><td colspan="2">グループ</td><td colspan="2">グループ</td></tr>
<tr><td>ノード</td><td>ノード</td><td>ノード</td><td>ノード</td><td>ノード</td><td>ノード</td><td>ノード</td></tr>
</table>

1.2.1. 管理者

AirManage 2 サーバの管理者には、システム管理者とテナント管理者の二種類があります。

1.2.1.1. システム管理者

システム管理者は、AirManage 2 サーバに、テナント管理者登録、テナント作成、ユニット登録、ノード作成ができます。本サービスでは、ぷらっとホーム株式会社がシステム管理者となります。

1.2.1.2. テナント管理者

テナント管理者は、その管理下にあるテナントに所属するノードに対して権限があります。システム管理者が、テナント管理者を新規登録することができます。

1.2.2. テナント

システム管理者がテナントを作成し、テナント管理者を指定します。テナント管理者は、管理下にあるテナントに所属するノードに対して権限があります。

1.2.3. グループ

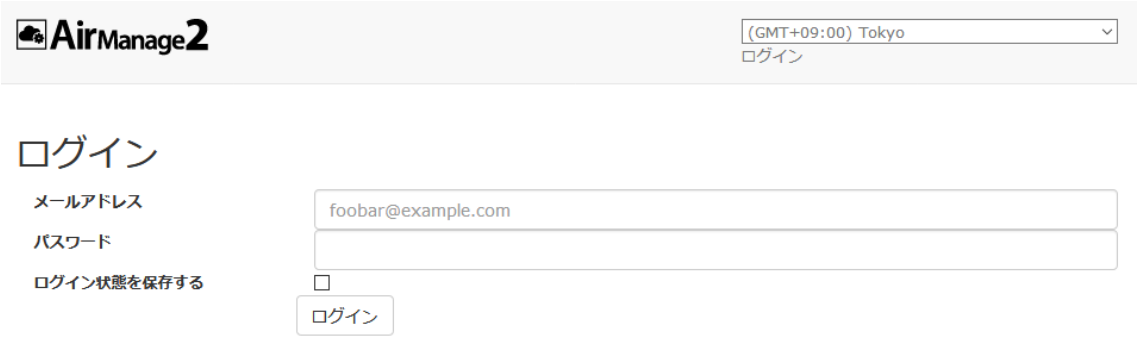
テナント管理者がグループを作成し、ノードを所属させることができます。ノードは、デフォルトで、DEFAULT グループに所属します。

2. 初期設定

本章では、AirManage 2 サーバの設定からノードの接続までを説明します。

2.1. ログイン(テナント管理者)

ブラウザで AirManage 2 サーバに接続します。

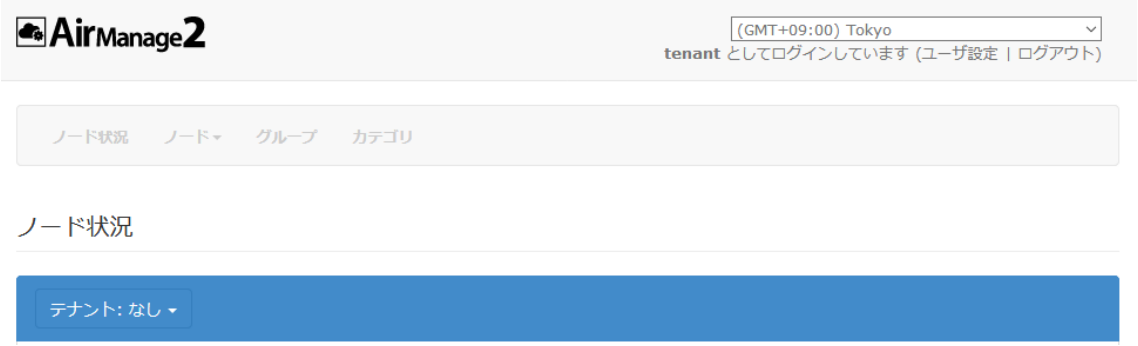


2.1.1. タイムゾーン

タイムゾーンは、画面右上のプルダウンメニューから選択します。

2.1.2. ログイン/ログアウト

ログインすると、ノード状況が表示されます。



ログアウトは、画面右上の[ログアウト]をクリックします。

2.1.3. ユーザの編集

テナント管理者の設定を変更するには、画面右上の[ユーザ設定]をクリックします。

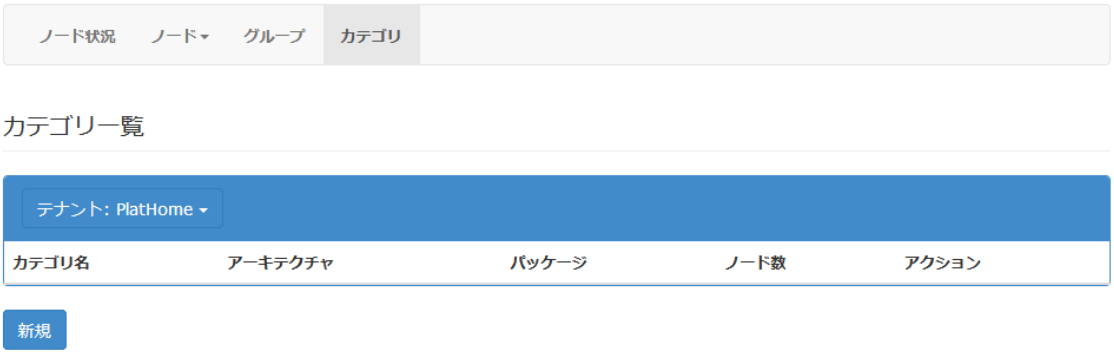
項目	内容
メールアドレス	メールアドレスです。ログイン時に入力します。
名前	テナント管理者の名前です。適当な文字列を入力します。
電話番号	連絡先の電話番号です。必要に応じて入力します。
アクセストークン	WebAPI のアクセストークンです。
2 段階認証	Google Authenticator で 2 段階認証を設定します。有効にする場合、[有効にする]ボタンを押します。2 段階認証設定画面で、QR コードを Google Authenticator で読み取り、コードを入力して、[認証する]ボタンを押します。
現在のパスワード	パスワードを変更する場合、現在のパスワードを入力します。
パスワード	パスワードを変更する場合、新しいパスワードを入力します。
パスワードの確認	確認のために、上の新しいパスワードを入力します。

終了するには、[アカウント詳細を更新]ボタンを押します。
パスワードを変更した場合、自動でログアウトされますので作業を継続する際には再度ログインを行ってください。

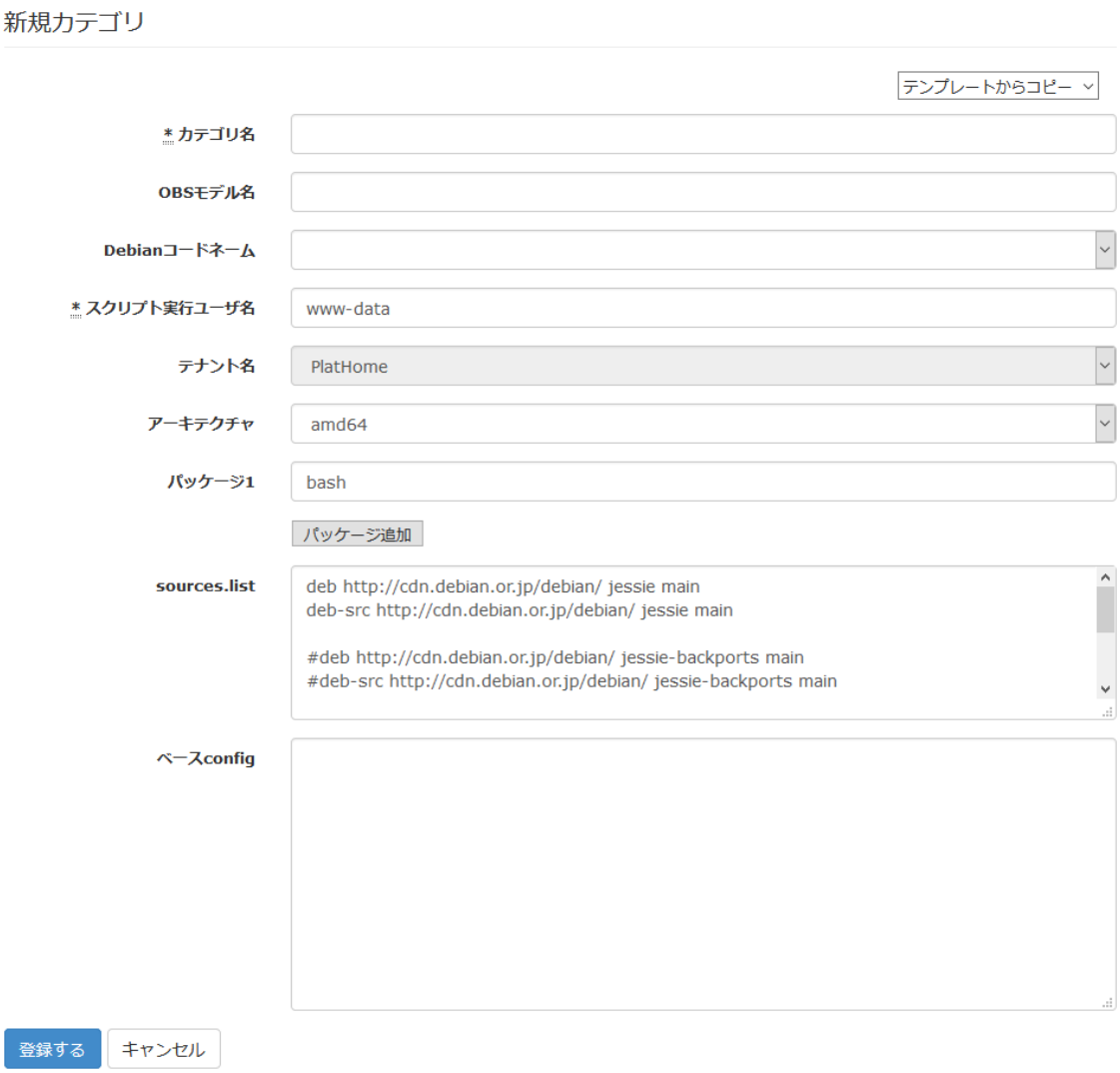
2.2. カテゴリ

ノードのパッケージのバージョンを管理するために、カテゴリを設定することができます。カテゴリが設定されていない場合、ノードからのアクセスが行われた際に自動でデフォルトのカテゴリがテナント用にコピーされますので、本カテゴリ編集は必須ではありません。カスタマイズをしたカテゴリを使用する場合のみ、本項目をご確認ください。

[カテゴリ]タブを選択します。



カテゴリを追加するには、[テナント: テナント名]ボタンにマウスカーソルを置いて表示されるテナントを選択して、[新規]ボタンを押します。



右上の[テンプレートからコピー]のセレクタにて対象のテンプレートを選択することにより、各フォームにデフォルトの値が入力されます。
カスタマイズする際の元データとしてご使用ください。

項目	内容
カテゴリ名	カテゴリの名前です。適当な文字列を入力します。
OBS モデル名	ノードのモデルのモデル名です。
Debian コードネーム	ノードのモデルの Debian コードネームです。
スクリプト実行ユーザ名	AirManage 2 サーバからノードに対してスクリプトを実行するときの実行ユーザ名です。ノードのモデル毎に決っています。OpenBlocks IoT Family 及び EasyBlocks は www-data です。

項目	内容
テナント名	カテゴリが所属するテナント名です。変更できません。
アーキテクチャ	ノードのモデルのアーキテクチャです。
パッケージ 1～8	バージョン管理を行う対象のパッケージ名を入力します。[パッケージ追加]ボタンを押して、8 個まで登録できます。
sources.list	更新するパッケージが置いてあるリポジトリを記載します。 追加の自作パッケージのリポジトリを追加する場合には、末尾に追記します。
ベース config	ノードの WEB UI の設定ファイルです。

終了するには、[登録する]ボタンを押します。

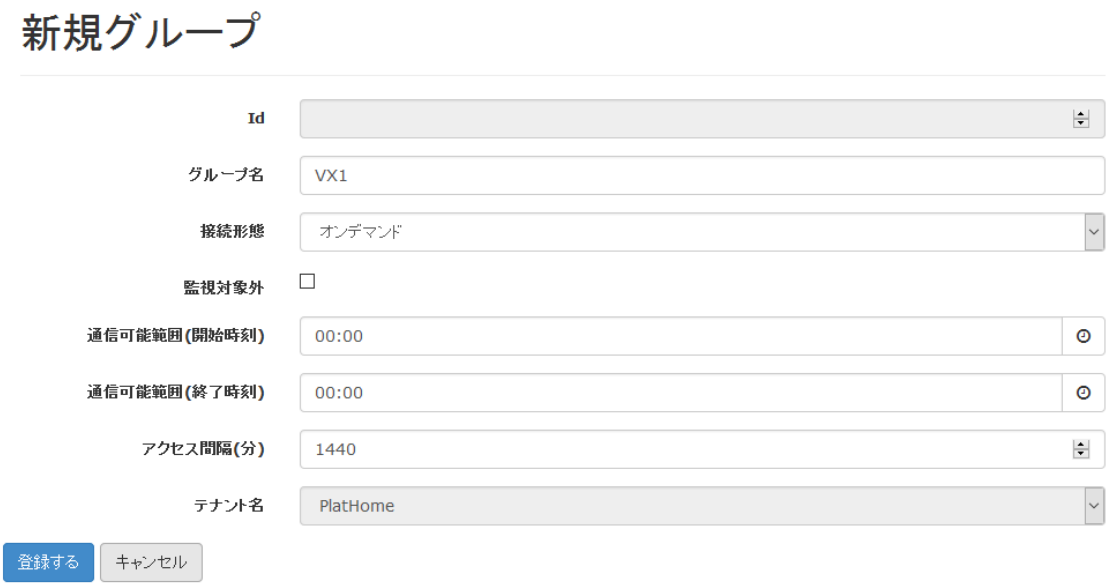
カテゴリの設定を変更するには、[編集]ボタンを押します。パッケージの最新版を確認するには、[最新版確認]ボタンを押します。カテゴリを削除するには、[削除]ボタンを押します。

カテゴリの設定を変更した場合、[最新版確認]ボタンを押して、パッケージの各パッケージ名のうしろの括弧内にバージョンが表示されていることを確認してください。パッケージ名のうしろに括弧が表示されない場合、設定のパッケージ名や sources.list などに誤りがあります。

2.3. グループ

[グループ]タブを選択します。
DEFAULT グループが存在します。このグループはユニットを適用したときに、グループがなければ作成されます。そのとき生成されたノードは、DEFAULT グループに属します。

グループを追加するには、[テナント: テナント名]ボタンにマウスカーソルを置いて表示されるテナントを選択して、[新規]ボタンを押します。



項目	内容
Id	グループの Id です。変更できません。
グループ名	グループの名前です。適当な文字列を入力します。
接続形態	常時かオンデマンドか選択します。
監視対象外	このグループの所属するノードを監視対象外にするには、チェックします。アラートメールが送信されません。
通信可能範囲(開始時刻)	オンデマンド接続時の通信可能な時間の開始時刻を入力します。(許容範囲: 00:00-23:59)
通信可能範囲(終了時刻)	オンデマンド接続時の通信可能な時間の終了時刻を入力します。(許容範囲: 00:00-23:59)
アクセス間隔(分)	オンデマンド接続時の通信可能範囲内での通信間隔を入力します。(許容範囲: 10-1440)
テナント名	グループが属するテナント名です。変更できません。

終了するには、[登録する]ボタンを押します。

グループの設定を変更するには、[編集]ボタンを押します。グループを削除するには、[削除]ボタンを押します。

2.4. ノード

[ノード]タブを選択します。

2.4.1. Config 以外の設定

ノードの所属するグループ、カテゴリ、パッケージのダウンロード方法と更新方法を設定します。設定するには、[編集]ボタンを押します。

項目	内容
ノード名	ノードの名前が表示されます。
グループ	ノードの所属するグループを選択します。
カテゴリ	ノードのカテゴリを選択します。
場所	ノードの設置場所です。空白のままか、適当な文字列を入力します。
メモ	ノードに関する覚え書きです。空白のままか、適当な文字列を入力します。
ダウンロード方法	パッケージのダウンロード方法を選択します。
ダウンロード時刻	ダウンロード方法に時間指定を選択した場合、パッケージをダウンロードする時刻を入力します。
更新方法	パッケージの更新方法を選択します。
更新時刻	更新方法に時間指定を選択した場合、パッケージを更新する時刻を入力します。

終了するには、[更新する]ボタンを押します。ノードを削除するには、[削除]ボタンを押します。

複数のノードを一括して設定することもできます。設定したいノードのチェックボックスで選択し、[一括編集]ボタンを押します。[一括編集]ウィンドウが開きます。設定内容は、上記と同じです。終了するには、[設定]ボタンを押します。
尚、カテゴリにはアーキテクチャーや FW 情報が含まれます。そのため、一括編集では異なるアーキテクチャーのノード(ex. OBSVX2 と OBSEX1 等)や FW ver.が異なるノード(ex. FW2 系のノードと FW3 系のノード)を同時に一括編集しないでください。

2.4.2. Config 管理

ノードの config 設定をします。設定するには、[Config 管理]ボタンを押します。

オリジナル世代の[編集]ボタンを押します。ノードの config ファイルを貼り付け、[更新する]ボタンを押します。

複数のノードを一括して設定することもできます。2.2 節で設定したベース config の全体または一部を上書きします。

注意

下記の設定をしないと、ノードには config が設定されません。

- カテゴリのベース config
- ノードのカテゴリ

1. ノードの JSON ファイルを用意します。書式は以下の通りです。

```
[
    {
        "node": "BX1_10",
        "config": {
```

```
                "network": {
                    "interface": {
                        "interfaces": {
                            "eth0": {
                                "mode": "static",
                                "address": "192.168.10.156",
                                "netmask": "255.255.255.0"
                            }
                        }
                    }
                }
            }
        },
        {
            "node": "BX1_11",
            "config": {
                "network": {
                    "interface": {
                        "interfaces": {
                            "eth0": {
                                "mode": "static",
                                "address": "192.168.10.157",
                                "netmask": "255.255.255.0"
                            }
                        }
                    }
                }
            }
        }
]
```

2. [ノード]タブにマウスカーソルを置くと、プルダウンメニューに[ノード config 一括設定]が表示されるので選択します。

3. [JSON ファイルを選択]ボタンを押して、JSON ファイルを選択します。
4. [JSON ファイルを読み込む]ボタンを押します。
   i. [JSON ファイル読み込み]ウィンドウが開きます。「登録終了」が表示されたら、[完了]ボタンを押します。

2.5. ノードの接続

以上で AirManage 2 サーバの設定は終了です。OpenBlocks IoT Family や EasyBlocks のユニットを AirManage 2 サーバに接続するには、各製品 WEB UI セットアップガイドまたはユーザーズガイドをご覧ください。

3. 運用

本章では、運用時に必要になる設定を説明します。2 章で説明した初期設定と重なる場合は、対応する節を参照します。

3.1. ユニットの交換

ユニットが故障のため修理してシリアル番号が変更された場合、交換前のシリアル番号と交換後のシリアル番号を AirManage 2 のサポート担当者に連絡します。

3.2. カテゴリ

[カテゴリ]タブを選択します。

カテゴリの追加、設定変更、削除については、2.2 節を参照してください。

3.3. グループ

[グループ]タブを選択します。

グループの追加、設定変更、削除については、2.3 節を参照してください。

ノードにアテンションがある場合、アテンションメールが送信されます。アテンションメールの送信を行いたくない場合には、監視対象外のグループに対象ノードを所属させることによってアテンションメールの送信が停止されます。

注意

編集しているグループにノードが所属している場合、[更新する]ボタンを押すと確認画面が表示されます。[はい]ボタンを押すと、所属しているノードは再起動します。

3.4. ノード

[ノード]タブを選択します。ノード画面が表示されます。プルダウンメニューからテナントとグループを選択してください。(複数のテナントの管理者の場合、テナント選択が可能です)

項目	内容
チェックボックス	複数のノードを一括処理するときに、チェックします。
有効	○：接続できます。－：接続できません。システム管理者に尋ねてください。
ノード名	ノード名です。
場所	ノードの編集で設定した場所です。
カテゴリ	ノードの編集で設定したカテゴリです。
パッケージ Ver.	ノードのパッケージ Ver.の一覧です。
有効期限	有効期限(00:00:00 UTC)がくるとノードが切断されます。
ステータス	AirManage 2 サーバとノードの稼動状態です。
次接続残時間(分)	接続形態がオンデマンドのとき表示されます。オンデマンド接続時の、次接続残時間です。
タウンロード	ダウンロード設定で設定したダウンロードの状態です。
更新	更新設定で設定した更新の状態です。
アクション	次のボタンが表示されます。[詳細]、[編集]、[サポートログ]、[Config 管理]、[Web アクセス](接続形態が常時のとき、または接続形態がオンデマンドでメンテナンスモードのときのみ)、[メンテナンスモード

項目	内容
	要求](接続形態がオンデマンドのときのみ)

3.4.1. ステータス

表示	内容
正常稼働中	ノードが正常に接続している状態です。接続形態がオンデマンドの場合、次回接続までの間も、そう表示されます。
正常稼働中(アテンション有)	ノードが正常に接続している状態です。ノードにアテンションがあることを示します。
未接続	ノードが接続していない状態です。ノードが停止している、ネットワークに異常がある場合などに表示されます。
未接続(アクセスなし)	ノードが一度も接続したことがありません。
更新あり(指示待ち)	新しいパッケージが存在します。必要があれば、ダウンロード設定と更新設定をしてください。
更新あり(DL 待機中)	新しいパッケージが存在します。ダウンロードの待機中です。
ダウンロード待ち	ダウンロードの待機中です。
ダウンロード実行中	ダウロードを実行中です。
ダウンロード失敗	ダウンロードに失敗しました。
アップデート指示待ち	更新設定をしてください。
アップデート待機中	更新の実行の待機中です。
アップデート適用中(実行待ち)	更新の実行の待機中です。まもなく実行されます。
アップデート適用中(再起動中)	更新の実行が終了し、OS の再起動中です。
アップデート適用失敗	更新の実行に失敗しました。

3.4.2. ダウンロード

表示	内容
停止	ダウンロードを実行しません。
時刻	ダウンロードが実行される時刻です。

3.4.3. 更新

表示	内容
停止	更新を実行しません。
ダウンロード直後	ダウンロードが終了したら更新を実行します。
時刻	更新を実行する時刻です。

3.4.4. 詳細

ノードに関する詳細情報を表示するには、[詳細]ボタンを押します。ノード情報表示画面が表示されます。

終了するには、[戻る]ボタンを押します。ノードの編集をするには、[編集]ボタンを押します。

3.4.4.1. 設定情報

項目	内容
Id	AirManage 2 サーバにおけるノードの識別番号です。
ノード名	ノード名です。
HW 名	ユニット管理で設定した HW 名です。
グループ	ノードの編集で設定したグループ名です。
接続形態	グループの編集で設定した接続形態です。
カテゴリ	ノードの編集で設定したカテゴリです。
場所	ノードの編集で設定した場所です。
メモ	ノードの編集で設定したメモです。
有効期限	有効期限です。
次回接続予定時刻	接続形態がオンデマンドのとき表示されます。次回接続予定時刻です。
次接続残時間(分)	接続形態がオンデマンドのとき表示されます。次回接続予定時刻までの時間です。
ダウンロード方法	ノードの編集で設定したダウンロード方法です。
ダウンロード時刻	ノードの編集で設定したダウンロード時刻です。
更新方法	ノードの編集で設定した更新方法です。
更新時刻	ノードの編集で設定した更新時刻です。

項目	内容
コンフィグ	Config 設定をするには、[Config 管理]ボタンを押します。

3.4.4.2. サーバ側情報

項目	内容
ノード制御用ポート番号	ノードが接続時に使用する TCP ポート番号です。
ウェブ接続用ポート番号	Web アクセスに使用する TCP ポート番号です。
接続状態	ノードの接続状態を表示します。
ステータス	AirManage 2 サーバが管理するノードの稼動状態を表示します。
待機アップロードファイル	待機アップロードファイルの有無です。
待機ジョブ	待機ジョブの有無です。

3.4.4.3. 取得情報

項目	内容
バージョン取得時刻	ノードのパッケージのバージョンを取得した最終時刻です。
最終接続確認時刻	ノードが接続したのを確認した最終時刻です。
パッケージ Ver.	ノードのパッケージ Ver.の一覧です。
タイムゾーン	ノードのタイムゾーンです。
アテンション	ノードのアテンションを表示します。
MAC アドレス	各ネットワークインタフェースの MAC アドレスです。
メモリ使用量(MB)	メモリの使用量です。
ディスク使用量(MB)	ディスクの使用量です。
IP アドレス	各ネットワークインタフェースの情報です。

3.4.5. 編集

編集については、2.4.1 節を参照してください。

3.4.6. サポートログ

弊社のサポート担当者からサポートログを取得するよう要請された場合、[サポートログ]ボタンを押します。

ノードから AirManage 2 サーバへサポートログを転送するために、[転送ジョブ予約]ボタンを押します。常時接続の場合は直後に、オンデマンド接続の場合は次接続時に転送します。

転送が終了すると、転送完了時刻、サイズ、[ダウンロード]ボタンが表示されます。サポートログをダウンロードするには、[ダウンロード]ボタンを押します。

終了するには、[ノード一覧へ戻る]ボタンを押します。

3.4.7. Config 管理

ノードの config を管理するには、[Config 管理]ボタンを押します。ノード Config 一覧画面が表示されます。

ノードの config が一覧されます。オリジナル世代は、2.4.2 節で設定した config です。ノードの WEB UI で設定を変更した場合、最新版を含めて 3 世代までバックアップします。表示するには、[表示]ボタンを押します。オリジナルとの差分を表示するには、[オリジナル

との差分]ボタンを押します。1 世代前との差分を表示するには、[1 世代前との差分]ボタンを押します。

オリジナルの config を編集し保存することによって、リコンフィグ用覧にチェックが入ります。

リコンフィグ用覧にチェックの入った世代の config が、リコンフィグで適用されます。

終了するには、[ノードへ戻る]ボタンを押します。

3.4.8. Web アクセス

接続形態が常時のとき、ノードの WEB UI を表示するには、[Web アクセス]ボタンを押します。

3.4.9. メンテナンスモード

接続形態がオンデマンドのとき、ノードの WEB UI を表示するためのモードです。

[メンテナンスモード要求]ボタンを押します。ノードが接続するまで、背景が黄色の[▶]ボタンと[メンテナンスモード終了]ボタンが表示されます。メンテナンスモード要求を中止するには、[メンテナンス終了]ボタンを押します。

ノードが接続すると、背景が黄色の[▶]ボタンが、背景が緑色の[Web アクセス]ボタンに換ってメンテナンスモードになります。ノードが接続したままになります。ノードの WEB UI を表示するには、[Web アクセス]ボタンを押します。メンテナンスモードを終了するには、[メンテナンス終了]ボタンを押します。

3.4.10. ダウンロード設定

ダウンロード設定するには、ノードをチェックボックスで選択して、[ダウンロード設定]ボタンを押します。ダウンロード画面でダウンロード方法(常時接続: 停止、今すぐ確認、時刻指定。オンデマンド接続: 停止、次回接続時)を選択して、[設定]ボタンを押します。

3.4.11. 更新設定

更新設定するには、ノードをチェックボックスで選択して、[更新設定]ボタンを押します。更新画面で更新方法(停止、ダウンロード直後、今すぐ更新、時刻指定)を選択して、[設定]ボタンを押します。

オンデマンド接続の場合、[更新設定]ボタンは表示されません。ダウンロード直後に更新します。

3.4.12. 一括編集

一括編集により、複数のノードの、グループ、カテゴリ、場所、メモ、ダウンロード方法、更新方法、を一括して編集できます。ノードをチェックボックスで選択して、[一括編集]ボタンを押します。一括編集画面で編集して、[設定]ボタンを押します。
尚、カテゴリにはアーキテクチャーや FW 情報が含まれます。そのため、一括編集では異なるアーキテクチャーのノード(ex. OBSVX2 と OBSEX1 等)や FW ver.が異なるノード(ex. FW2 系のノードと FW3 系のノード)を同時に一括編集しないでください。

3.4.13. ファイルアップロード

本機能は特定のファイル名規則での tar+gz 形式のファイルを AirManage 2 サーバへアップロードすることによって、アップロード対象のノードに対してファイルが配信されます。この tar+gz ファイル内の特定のシェルスクリプトを実行できる機能となります。
本機能を使用方法は、ノードをチェックボックスで選択し、[ファイルアップロード]ボタンを押します。ファイルアップロード画面にて、ファイルを選択し、[アップロード]ボタンを押します。常時接続のノードに対しては、即時にファイルアップロードが実行されます。尚、未接続状態のノードに対しては、接続後にファイルアップロードされます。また、オンデマンド接続では次接続時にファイルアップロードが実行されます。

■アップロードファイル名規則
"am*<MODE>*_*<TYPE>*_*<ARCH>*"または"am*<MODE>*_*<TYPE>*_all"から始まるファイル名。

■*<MODE>*変数説明
アップロードしたファイル内のシェルスクリプト実行タイミングを<MODE>変数によって制御します。

*<MODE>*変数内容	シェルスクリプト実行タイミング
boot	起動時に処理する場合
inst	起動中(ファイル配信直後)に処理する場合
halt	終了時に処理する場合

■*<TYPE>*変数説明

間違ったアップロードファイルを処理しないようにノード毎にできる*<TYPE>*を可変としています。デフォルト値は”none"です。ノード側の/etc/obsiot/air_model_type ファイルにTYPE 名を定義することによって、*<TYPE>*を変更することができます。

※ファイル内に 1 行で*<TYPE>*としたい文字列のみを記載してください。尚、英数字のみサポートなります。

■*<ARCH>*変数説明

CPU アーキテクチャー依存の処理が対応の為、ファイル名に以下の変数を設定し制御します。

*<ARCH>*変数内容	対応モデル
amd64	OBSVX1 , OBSVX2
i386	OBSBX1, OBSBX3, OBSBX5, OBSEX1, OBSEX1G, OBSBX0

■アップロードファイルフォーマット

tar+gz による圧縮ファイルとなります。

■アップロードファイル内データ構成

・実行シェルスクリプト(必須ファイル / 各処理フェーズにて必要ファイルが異なります。)

boot.sh： 起動時に処理する場合に必用となります。

inst.sh： 起動中に処理する場合に必用となります。

halt.sh： 終了時に処理する場合に必用となります。

・その他(必要に応じてとなります。)

実行シェルスクリプト内等で処理に必要なファイルを用意してください。

※起動時処理用サンプルファイルデータ構成

```
# tar tvf   amboot_none_amd64.tgz
-rw-r--r-- root/root          350 2018-11-22 09:51 ./boot.sh
-rw-r--r-- root/root            0 2018-11-22 09:47 ./data1
```

-rw-r--r-- root/root	0 2018-11-22 09:47 ./data2
-rw-r--r-- root/root	0 2018-11-22 09:47 ./data3
-rw-r--r-- root/root	0 2018-11-22 09:47 ./data4
-rw-r--r-- root/root	0 2018-11-22 09:47 ./data5

■アップロード時展開先ディレクトリ
該当のシェルスクリプト処理が終了した場合、作業ディレクトリは削除されます。そのため、データ等を残す場合にはシェルスクリプト内でコピー等を実施してください。

・起動時処理時ディレクトリ
/var/tmp/BOOT_DIR

・起動中処理時ディレクトリ
/var/tmp/INST_DIR

・終了時処理時ディレクトリ
/var/tmp/HALT_DIR

3.4.14. リコンフィグ

リコンフィグ機能は以下の 3 種類の処理が行えます。

項目	処理内容
リコネクション	AirManage2 サーバ側のコンフィグは反映せずに再接続を行います。
リコンフィグ(ネットワーク設定除く)	AirManage2 サーバ側のネットワーク部以外のコンフィグをノードへ適用し再接続を行います。 尚、本機能は以下の FW バージョン以降使用可能です。 ■OpenBlocks IoT Family の場合 ・FW3 系の場合、FW3.3.3 から使用可能となります。 ・FW2 系の場合、FW2.1.8 から使用可能となります。 ■EasyBlocks の場合 ・対応製品の初期バージョンから使用可能です。
リコンフィグ	AirManage2 サーバ側のコンフィグをノードへ適用し再接続を行います。

リコンフィグを実行するには、ノードをチェックボックスで選択して、[リコンフィグ]ボタンを押します。リゴンフィグ実行画面で、タイプ(リコネクション、リコンフィグ(ネットワーク設定除く)、リコンフィグ)を選択し、 [設定]ボタンを押します。常時接続では、ただちにリコンフィグが実行されます。オンデマンド接続では、次接続時にリコンフィグが実行されます。
尚、サーバ URL が記入されている場合には、記載内容の URL の AirManage 2 サーバへとリコンフィグ処理が行われます。未記入の場合には接続中の AirManage 2 サーバに対してリコンフィグ処理が行われます。

ここで配布される config は、3.4.7 節で説明した、初期設定用覧にチェックのついた世代です。
本機能は弊社からサーバ切り替えのご案内時に用いることがありますので、ご案内があった場合には速やかに適用をお願いいたします。

3.5. ノード状況

[ノード状況]タブを選択します。

各項目はリンクになっています。クリックするとその項目でフィルタしたノード画面に遷移します。フィルタを解除するには、[×]ボタンを押します。

項目	値
ノード	テナント内のノードの個数です。
監視対象外ノード	監視対象外にチェックの入ったグループに所属するノードの個数です。
監視対象ノード	テナント内のノードの個数から監視対象外ノード数を引いたノードの個数です。
接続不能ノード	AirManage 2 サーバに接続していないノードの個数です。
接続ノード	AirManage 2 サーバに接続しているノードの個数です。オンデマンド接続のノードで、接続を待機している場合は、含まれません。
ダウンロード待ち	ダウンロード待ちしているノードの個数です。
ダウンロード中	ダウンロード中のノードの個数です。
ダウンロード失敗	ダウンロードを失敗したノードの個数です。
アップデート指示待ち	アップデート指示待ちのノードの個数です。
アップデート待ち	アップデート待ちのノードの個数です。

項目	値
アップデート実行中	アップデート実行中のノードの個数です。
アップデート失敗	アップデートを失敗したノードの個数です。
アテンションあり	アテンションのあるノードの個数です。

4. WebAPI

本章は、AirManage 2 サーバの WebAPI について説明します。テナント管理者が実行可能な API の一覧となります。

4.1. テナントサマリ確認 API

自分が管理するテナント内のノードのサマリを確認できます。

4.1.1. リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/summary

4.1.2. リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

JSON キー	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
group_id	グループの Id。グループ画面で確認します。指定しないとテナントに所属する監視対象のノードの情報を取得します。
token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。

4.1.3. レスポンス

レスポンスは JSON 形式です。内容は、ノード状況画面で表示される情報と同じです。詳しくは、3.5 節をご覧ください。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果
summary	JSON オブジェクト
total	全ノード数
unmonitored	監視対象外ノード数
monitored	監視対象ノード数
inactive	接続不能ノード数
active	接続ノード数
download_queued	ダウンロード待ち
download_running	ダウンロード中
download_finished	アップデート指示待ち
download_failed	ダウンロード失敗
upgrade_queued	アップデート待ち
upgrade_running	アップデート実行中
upgrade_failed	アップデート失敗
attention	アテンションあり

4.1.4.例

cURL コマンドでのアクセス例です。

- グループ Id を指定しない場合

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバの FQDN>/api/v1/nodes/summary' -d '{ "tenant_code": "テナント記号", "token": "アクセストークン"}'
```

- グループ Id を指定した場合

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバのホスト名>/api/v1/nodes/summary' -d '{ "tenant_code": "テナント記号", "group_id": グル
```

```
ープ Id, "token": "アクセストークン"}'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
  "result": "success",
  "summary": {
    "total": 2,
    "unmonitored": 0,
    "monitored": 2,
    "inactive": 0,
    "active": 2,
    "download_queued": 0,
    "download_running": 0,
    "download_finished": 0,
    "download_failed": 0,
    "upgrade_queued": 0,
    "upgrade_running": 0,
    "upgrade_failed": 0,
    "attention": 0
  }
}
```

4.2. ノード簡易情報確認 API

自分が管理するテナント内のノードの簡易情報を確認できます。

4.2.1. リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/list

4.2.2. リクエスト

リクエストメソッドは GET です。または、リクエストデータ形式はクエリです。

パラメータ	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。

4.2.3. レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果
nodes	JSON オブジェクト配列
id	ノードのユニーク ID
location	場所情報(未設定の場合、null)
ping_at	最終接続確認時刻
connected	接続状態
ondemand_maintenance	オンデマンド接続時におけるメンテナンスモード嬢状況
group_name	グループ名

4.2.4. 例

cURL コマンドでのアクセス例です。

```
$ curl -s -X GET -k 'https://<AirManage 2 サーバの FQDN>/
api/v1/nodes/list?tenant_code=＜テナント記号＞&token=＜アクセストークン＞
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
  "result": "success",
  "nodes": [
    {
      "id": 29,
      "name": "obssvr_stretch",
      "location": null,
      "ping_at": "2019-05-27T06:55:35.204Z",
      "connected": true,
      "ondemand_maintenance": null,
      "group_name": "DEFAULT"
    },
    {
      "id": 26,
      "name": "obsex_stretch",
```

```
        "location": "",
        "ping_at": "2019-05-27T06:54:54.668Z",
        "connected": false,
        "ondemand_maintenance": "request",
        "group_name": "EDISON"
      },
      {
        "id": 4,
        "name": "obsex_jessie",
        "location": "",
        "ping_at": "2019-05-27T06:55:26.879Z",
        "connected": false,
        "ondemand_maintenance": null,
        "group_name": "EDISON"
      },
      {
        "id": 27,
        "name": "obsvx_stretch",
        "location": null,
        "ping_at": "2019-05-27T06:56:04.260Z",
        "connected": true,
        "ondemand_maintenance": null,
        "group_name": "DEFAULT"
      }
    ]
}
```


4.3. ノード詳細情報確認 API

自分が管理するテナント内のノードの詳細情報を確認できます。

4.3.1. リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/node_status

4.3.2. リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

キー	内容
tenant_code	テナント記号。テナント一覧画面で確認します。

token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。
node_id または node_name	詳細情報を確認したいテナント内のノードのユニーク ID またはノード名。

4.3.3. レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果
nodeinfo	JSON オブジェクト
id	ノードのユニーク ID。
group_id	ノードが所属するグループの ID。
name	AirManage 上で登録されているノード名。
hwname	AirManage 上で登録されている HW 名。
status	ノードのステータス情報。
ping_at	最終通信確認時刻。
location	場所情報。
connected	接続状態。
attention	アテンション情報。
uploadfile_id	ファイルアップロード機能のアップロード対象とするファイル ID。
expired_at	有効期限。
active	AirManage サービスの有効・無効状態。
jobs	AirManage での待機ジョブ一覧。

4.3.4. 例

cURL コマンドでのアクセス例です。

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバの FQDN>/api/v1/nodes/node_status' -d '{ "tenant_code": "テナント記号", "token": "アクセス
```

```
トークン" , “node_id” : “ノードのユニーク ID” }'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。


```
{
   "result": "success",
   "nodeinfo": {
      "id": 27,
      "group_id": 1,
      "name": "obsvx_stretch",
      "hwname": "I1J00035",
      "status": "old_revision",
      "ping_at": "2019-05-27T07:11:22.255Z",
      "location": null,
      "connected": true,
      "attention": null,
      "uploadfile_id": null,
      "expired_at": null,
      "active": true,
      "jobs": []
   }
}
```


4.4. テナント詳細確認 API

自分が管理するテナント内のノード全体の詳細を確認できます。

4.4.1.リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/tenant_status

4.4.2.リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

JSON キー	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
group_id	グループの Id。グループ画面で確認します。指定しない場合、テナントに所属する全体のノードの情報を取得します。

token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。

4.4.3.レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果
nodesinfo	JSON オブジェクト配列
id	ノードのユニーク ID。
group_id	ノードが所属するグループの ID。
name	AirManage 上で登録されているノード名。
hwname	AirManage 上で登録されている HW 名。
status	ノードのステータス情報。
ping_at	最終通信確認時刻。
location	場所情報。
connected	接続状態。
attention	アテンション情報。
uploadfile_id	ファイルアップロード機能のアップロード対象とするファイル ID。
expired_at	有効期限。
active	AirManage サービスの有効・無効状態。
jobs	AirManage での待機ジョブ一覧。

4.4.4.例

cURL コマンドでのアクセス例です。

- グループ Id を指定しない場合

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバの FQDN>/api/v1/nodes/tenant_status' -d '{ "tenant_code": "テナント記号", "token": "アクセストークン"}'
```

- グループ Id を指定した場合

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバのホスト名>/api/v1/nodes/tenant_status' -d '{ "tenant_code": "テナント記号", "group_id": グループ Id, "token": "アクセストークン"}'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。


```
{
  "result": "success",
  "nodesinfo": [
    {
      "id": 4,
      "group_id": 5,
      "name": "obsex_jessie",
      "hwname": "FAE00857",
      "status": "old_revision",
      "ping_at": "2019-05-27T07:16:13.007Z",
      "location": "",
      "connected": false,
      "attention": null,
      "uploadfile_id": null,
      "expired_at": null,
      "active": true,
      "jobs": []
    },
    {
      "id": 26,
      "group_id": 5,
      "name": "obsex_stretch",
      "hwname": "G3E00015",
      "status": "old_revision",
      "ping_at": "2019-05-27T07:20:12.292Z",
      "location": "",
      "connected": false,
      "attention": null,
      "uploadfile_id": null,
      "expired_at": null,
      "active": true,
      "jobs": []
    }
  ]
}
```


4.5. ファイルアップロード API

自分が管理するテナント内のノードに配信するためのファイルをアップロードできます。
尚、本 API ではノードへのファイル配信自体は行われません。

4.5.1.リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/fileupload

4.5.2.リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式はマルチパートフォームです。

パラメータ	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
group_id	ファイルをアップロード対象とするグループの Id。グループ画面で確認します。
token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。
uploadfile	アップロード対象のファイル。

4.5.3.レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果
file_id	アップロード時に割り振られるファイル ID。
filename	アップロード時のファイル名です。

4.5.4.例

cURL コマンドでのアクセス例です。

```
$ curl -s -X POST -k -H "Content-type: multipart/form-data" 'https://<AirManage 2 サー
```

```
バの FQDN>/api/v1/nodes/fileupload’￥
-F uploadfile=@＜アップロードファイル＞ ￥
-F token=＜アクセストークン＞ ￥
-F tenant_code=＜テナント記号＞ ￥
-F group_id=＜グループ ID＞
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
   "result": "success",
   "file_id": 34,
   "filename": "test.tgz"
}
```

4.6. ファイルアップロードジョブ API

自分が管理するテナント内のノードへのファイル配信ジョブの登録を行います

4.6.1.リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/filejob

4.6.2.リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

パラメータ	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
group_id	ファイルをアップロード対象とするグループの Id。グループ画面で確認します。※ファイルを AirManage にアップロード時のグループ ID と同一である必要があります。
node_id または node_name	ファイル配信対象のテナント内のノードのユニーク ID またはノード名。

token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。
file_id	ファイルアップロード API のレスポンスの ID。

4.6.3. レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果

4.6.4. 例

cURL コマンドでのアクセス例です。

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバの FQDN>/api/v1/nodes/filejob' -d '{ "tenant_code": "テナント記号", "token": "アクセストークン" , "group_id"： グループ ID , "node_id"： ノード ID , "file_id"  ： ファイル ID }'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
   "result": "success"
}
```

4.7. ファイルアップロードジョブ取消 API

自分が管理するテナント内のノードへのファイル配信ジョブの取り消しを行います

4.7.1. リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/delfilejob

4.7.2.リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

パラメータ	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
node_id または node_name	ファイル配信対象のテナント内のノードのユニーク ID またはノード名。
token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。

4.7.3.レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果

4.7.4.例

cURL コマンドでのアクセス例です。

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバの FQDN>/api/v1/nodes/delfilejob' -d '{ "tenant_code": "テナント記号", "token": "アクセストークン" , "node_id" : ノード ID }'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
   "result": "success",
}
```

4.8. ノード全ジョブ取消 API

自分が管理するテナント内のノードが実施処理予定のジョブの取り消しを行います

4.8.1.リクエスト URL

https://<AirManage 2 サーバの FQDN>/api/v1/nodes/clear_job

4.8.2.リクエスト

リクエストメソッドは POST です。または、リクエストデータ形式は JSON です。

パラメータ	内容
tenant_code	テナント記号。テナント一覧画面で確認します。
node_id または node_name	全ジョブを取り消す対象のテナント内のノードのユニーク ID またはノード名。
token	アクセストークン。テナント管理者のユーザの設定画面にて確認します。

4.8.3.レスポンス

レスポンスは JSON 形式です。以下は正常時のレスポンスの JSON キーとなります。

JSON キー	内容
result	実行結果

4.8.4.例

cURL コマンドでのアクセス例です。

```
$ curl -s -X POST -k -H "Content-type: application/json" 'https://<AirManage 2 サーバのFQDN>/api/v1/nodes/clear_job' -d '{ "tenant_code": "テナント記号", "token": "アクセストークン" , "node_id" : ノード ID }'
```

上記のコマンドを実行した場合の出力例(jq コマンドで整形)です。

```
{
   "result": "success",
   "message": "Exec Clear Jobs."
}
```

ぷらっとホーム株式会社
〒102-0073 東京都千代田区九段北 4-1-3 日本ビルディング九段別館 3F