

DEXPF API仕様書

Version 1.1-08

2020年6月29日
ぷらっとホーム株式会社

目次

DEXPF API全般	6
DEXPF APIでのJSONデータ	6
DEXPF APIでのID	7
JSONパラメータのリファレンス	8
オーナー関係のパラメータ	8
アプリケーション関係のパラメータ	8
デバイス関係のパラメータ	9
チャンネル関係のパラメータ	9
その他のパラメータ	10
DEXPF API のレスポンスヘッダー	10
トークンの扱い	10
DEXPF API 一覧	11
「curlでの例」についての説明	11
Accounts	11
POST /api/v1/Accounts	11
概要	11
パラメータ	11
レスポンス	12
GET /api/v1/Accounts	12
概要	12
パラメータ	12
レスポンス	12
GET /api/v1/Accounts/{id}	13
概要	13
パラメータ	13
レスポンス	13
PUT /api/v1/Accounts	14
概要	14
パラメータ	14
レスポンス	14
DELETE /api/v1/Accounts/{id}	15
概要	15
パラメータ	15
レスポンス	15
Authorizations	15
POST /api/v1/Authorizations/Token	15
概要	15

パラメータ	15
レスポンス	16
DeviceGroups	16
POST /api/v1/DeviceGroups	16
概要	16
パラメータ	16
レスポンス	17
GET /api/v1/DeviceGroups	17
概要	17
パラメータ	17
レスポンス	17
GET /api/v1/DeviceGroups/{id}	18
概要	18
パラメータ	18
レスポンス	18
DELETE /api/v1/DeviceGroups/{id}	18
概要	18
パラメータ	18
レスポンス	19
Devices	19
POST /api/v1/Devices	19
概要	19
パラメータ	19
レスポンス	20
GET /api/v1/Devices	21
概要	21
パラメータ	21
レスポンス	21
GET /api/v1/Devices/{id}	21
概要	21
パラメータ	21
レスポンス	22
DELETE /api/v1/Devices/{id}	22
概要	22
パラメータ	22
レスポンス	22
AppGroups	23
POST /api/v1/AppGroups	23
概要	23
パラメータ	23
レスポンス	24
GET /api/v1/AppGroups	24

概要	24
パラメータ	24
レスポンス	24
GET /api/v1/AppGroups/{id}	25
概要	25
パラメータ	25
レスポンス	25
DELETE /api/v1/AppGroups/{id}	25
概要	25
パラメータ	25
レスポンス	26
Apps	26
POST /api/v1/Apps	26
概要	26
パラメータ	26
レスポンス	27
GET /api/v1/Apps	27
概要	27
パラメータ	27
レスポンス	27
GET /api/v1/Apps/{id}	28
概要	28
パラメータ	28
レスポンス	28
DELETE /api/v1/Apps/{id}	29
概要	29
パラメータ	29
レスポンス	29
Channels	29
POST /api/v1/Channels	30
概要	30
パラメータ	30
レスポンス	30
GET /api/v1/Channels	31
概要	31
パラメータ	31
レスポンス	31
GET /api/v1/Channels/{id}	32
概要	32
パラメータ	32
レスポンス	32
PUT /api/v1/Channels/{id}	32

概要	32
パラメータ	32
レスポンス	33
DELETE /api/v1/Channels/{id}	33
概要	33
パラメータ	33
レスポンス	33
Version	34
GET /api/v1/Version	34
概要	34
パラメータ	34
レスポンス	34
Admin	34
POST /api/v1/Admin/Setup	34
概要	34
パラメータ	34
レスポンス	35
改訂履歴	35

DEXPF API全般

はじめにDEXPF APIの全般について述べる。以降の説明で単にAPIとある場合は、DEXPF APIを示す。

- APIは標準的なRESTインターフェースで構成されている
- APIではリソースに従ったURLが用意されており、必要に応じてJSON形式のパラメータを受け付ける
 - APIにJSONのパラメータを送る場合、HTTPヘッダとして
Content-Type: application/json
を指定する。API一覧の説明では省略しているため注意する。
- レスポンスでは、標準的なHTTP応答コードを返すと共に、JSONデータを応答する
- リクエストにエラーがあった場合、パラメータエラー(ステータスコード 400)に関しては、JSONデータでメッセージが返る。APIが用意していないURLをアクセスした場合等、DEXPFが想定していないエラーが発生した場合は、HTTP応答コードは返すがボディは空であったり、テキスト形式となることもある。

DEXPF APIでのJSONデータ

APIが受け付けるJSONデータについての注意事項は次の通りである。

- JSONフォーマットとして正しいものであること
- JSON形式のキーは大文字、小文字を区別しない
 - たとえば "username"であれば、"username"、"USERNAME"、"userName"、
のいずれであっても、同じ"username"として取り扱われる。
- 同じキーが複数存在した場合は、最後に設定したキーが有効となる
- APIが使用しないキーと値のペアが存在した場合は無視される

DEXPF APIでのID

DEXPFのAPIでは、デバイスやアプリケーション等をIDで識別する。APIで使用するIDの一覧は次の通りである。

IDの名称	説明
オーナーID	オーナーを識別するID オーナー登録時にDEXPFにより設定される 16進数8桁の文字列
アプリケーション グループID	アプリケーショングループを識別するID オーナーが割り当てる 16進数8桁の文字列で範囲は a0000001 から affffffe まで
アプリケーションID	アプリケーションを識別するID オーナーが割り当てる 16進数8桁の文字列で範囲は f0000001 から fffffffe まで
デバイス グループID	デバイスグループを識別するID オーナーが割り当てる 16進数8桁の文字列で範囲は 90000001 から 9fffffff まで
デバイスID	デバイスを識別するID オーナーが割り当てる 16進数8桁の文字列で範囲は 00000001 から effffffe まで
チャンネルID	チャンネルを識別するID チャンネル作成時にDEXPFにより設定される 16進数8桁の文字列
システムID	DEXPFそのものを識別するID 初期設定時にDEXPFにより設定される 16進数8桁の文字列 現時点でこのIDを利用することは無い

JSONパラメータのリファレンス

オーナー関係のパラメータ

キー	説明
username	オーナーのユーザー名 6文字以上100文字以下のメールアドレス形式を利用する APIはメールアドレスであるかどうかをチェックしていないため、メールアドレス形式に適合しないものでもエラーとはならず受け付ける
password	オーナーのパスワード 6文字以上100文字以下のASCII文字列を使用する
ownerId	オーナーID
devOwnId	オーナーIDをデバイス側のオーナーとして識別する場合に使用する
appOwnId	オーナーIDをアプリケーション側のオーナーとして識別する場合に使用する
token	APIをリクエストする際に必要なトークン JWT Tokenで現バージョンでは2KB以下の文字列となる

アプリケーション関係のパラメータ

キー	説明
appGrId	アプリケーショングループID
appGrIds	appGrIdの配列
appId	アプリケーションID
appIds	appIdの配列
url	アプリケーション登録時に、アプリケーションのURL、つまりDEXPFからのHTTP POSTのデータを受け取るURLを指定する
headers	アプリケーション登録時に、アプリケーションにデータをPOSTするときのHTTPヘッダーを配列形式で指定する APIでは設定された文字列をチェックせず、そのままアプリケーションへのデータ送信に設定する、アプリケーションが受け付けられないような形式のものは指定した場合、エラーとなる

デバイス関係のパラメータ

キー	説明
devGrId	デバイスグループID
devGrIds	devGrIdの配列
devId	デバイスID
devIds	devIdの配列
hostname	デバイスが接続するAzure IoT Hubのホスト名
hubId	Azure IoT Hub接続に各デバイスが使用するデバイスID (DEXPFのデバイスIDとは別のもの)
key1	Azure IoT Hub接続時のプライマリ接続文字列
key2	Azure IoT Hub接続時のセカンダリ接続文字列
devices	デバイス登録時のレスポンスに含まれるJSONの配列 それぞれのJSONには devId と hubId、key1、key2 の4つのパラメータが含まれる

チャネル関係のパラメータ

キー	説明
chanId	チャネルID
chanIds	devGrIdの配列
devSide	チャネルのデバイス側の状態を示す "enable"であればチャネルのデバイス側が有効 "disable"であればチャネルのデバイス側が無効
appSide	チャネルのアプリケーション側の状態を示す "enable"であればチャネルのアプリケーション側が有効 "disable"であればチャネルのアプリケーション側が無効

その他のパラメータ

キー	説明
systemId	システムID
version	DEXPFのバージョンを示す文字列
status	必要に応じて "Ok" または "Error" が設定される
message	statusが "Error" の場合に、その簡単な説明が設定される

DEXPF API のレスポンスヘッダー

DEXPF APIのリクエストに対するレスポンスヘッダーには以下のものがある。必ずしも全てが含まれるわけではなくリクエストによって含まれ無いものもある。

- www-authenticate ヘッダ
- content-type ヘッダ
- content-length ヘッダ
- date ヘッダ

これ以外のヘッダについては、APIをアクセスする側は無視すること。

トークンの扱い

Adminやオーナーの権限が必要なAPIをアクセスする場合はAuthorizationヘッダで Bearer トークンを指定する。

- トークンには期限があり、有効期限は生成後1時間である
- トークンが有効期限を過ぎた場合は、再度生成する必要がある

トークンが誤っている場合は、次のようなレスポンスとなる

Code	説明
401	<p>www-authenticateヘッダで状況を示し、ボディは空となる 次の例は、トークンの有効期限を過ぎた場合のレスポンス</p> <hr/> <pre>www-authenticate: Bearer error="invalid_token", error_description="The token is expired" content-length: 0</pre>

DEXPF API 一覧

APIの一覧を以下に示す

「curlでの例」についての説明

説明ではcurlコマンドを使ったAPIを利用する場合の例を記述してある。

- Linux等のUNIX系OSのコマンドラインで使用する場合を想定している
- APIのホスト名は“api.dexpf.jp”としているが、実在しない仮想のホスト名を指定している
- 指定するパラメータについては例として適当なものを指定している

AuthorizationヘッダでBearerトークンを指定する部分については、トークンの表現を次の表のように省略して表記している。

<code>\${ADMIN_TOKEN}</code>	Adminアカウントのトークン
<code>\${OWNER_TOKEN}</code>	オーナーアカウントのトークン
<code>\${TOKEN}</code>	Adminアカウント、オーナーアカウントのいずれのトークンでも受け付けるもの

Accounts

POST /api/v1/Accounts

概要

DEXPFを利用する各オーナーを作成する。

パラメータ

Authorization ヘッダ	Bearer としてAdminアカウントのトークンを設定する
リクエスト ボディ	ユーザー名とパスワードを指定する <pre>{ "username": "string", "password": "string" }</pre>
curlでの例	<pre>curl --header "Authorization: Bearer \${ADMIN_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Accounts << EOT</pre>

	<pre>{ "username": "user@example.jp", "password": "wAuARgRKfD22vTL6pg98hEZjDBKZVi8c" } EOT</pre>
--	--

レスポンス

Code	説明
201	<p>オーナーの作成が成功し、レスポンスボディはownerIdが返る</p> <pre>{ "ownerId": "string" }</pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/Accounts

概要

Adminアカウントが実行した場合は、登録済オーナーIDの一覧を得る。一般のオーナーアカウントの場合は、GET /api/v1/Accounts/{id} でそのオーナーIDを指定した場合と同じ結果となる。

パラメータ

Authorization ヘッダ	Bearer としてトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${TOKEN}" \ https://api.dexpf.jp/api/v1/Accounts

レスポンス

Code	説明
200	<p>Adminアカウントが実行した場合、ownerIdの一覧が返る</p> <pre>{ "ownerId": ["string", "string", ] }</pre>

	<pre> } オーナーアカウントが実行した場合は、そのusernameとオーナーIDが返る { "id": "string", "username": "string" } </pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre> { "status": "Error", "message": "string" } </pre>

GET /api/v1/Accounts/{id}

概要

指定した ownerId の詳細を調べる。URLの{id}部分にはオーナーIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてAdminアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${TOKEN}" \ https://api.dexpf.jp/api/v1/Accounts/abcdef01

レスポンス

Code	説明
200	<p>ボディにownerIdの詳細が返る</p> <pre> { "id": "string", "username": "string" } </pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre> { "status": "Error", "message": "string" } </pre>

PUT /api/v1/Accounts

概要

パスワードを再設定する。Admin アカウントの権限で実行した場合は、オーナーアカウントのパスワードを再設定できる。

パラメータ

Authorization ヘッダ	Bearer としてトークンを設定する
リクエスト ボディ	ユーザー名と新しいパスワードを指定する <pre>{ "username": "string", "password": "string" }</pre>
curlでの例	<pre>curl --request PUT \ --header "Authorization: Bearer \${TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Accounts << EOT { "username": "user@example.jp", "password": "JRRBjKF3jzkDdaT274MeKpuH6ABDPU" } EOT</pre>

レスポンス

Code	説明
200	パスワードが変更できた <pre>{ "status": "Ok" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/Accounts/{id}

概要

登録済オーナーを削除する。URLの{id}部分にはオーナーIDを指定する。

【注意】オーナーがアプリケーショングループやデバイスグループを登録済である場合はオーナーの削除はできない。

パラメータ

Authorization ヘッダ	Bearer としてAdminアカウントのトークンを設定する
curlでの例	curl --request DELETE \ --header "Authorization: Bearer \${ADMIN_TOKEN}" \ https://api.dexpf.jp/api/v1/Accounts/abcdef01

レスポンス

Code	説明
200	ステータスが返る { "status": "Ok" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

Authorizations

POST /api/v1/Authorizations/Token

概要

Adminや各オーナーのトークンを生成する。トークンは生成してから1時間有効である。

パラメータ

リクエスト ボディ	アカウント作成時に使用したユーザー名とパスワードを指定する { "username": "string", "password": "string" }
--------------	--

	} }
curlでの例	curl --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Authorizations/Token << EOT { "username": "user@example.jp", "password": "wAuARgRKfD22vTL6pg98hEZjDBKZVi8c" } EOT

レスポンス

Code	説明
201	tokenの作成が成功 { "token": "string" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

DeviceGroups

POST /api/v1/DeviceGroups

概要

指定したIDのデバイスグループを作成する

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
リクエスト ボディ	デバイスグループIDを指定する { "devGrId": "string" }
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ --header "Content-Type: application/json" \

	<pre>--data @- \ https://api.dexpf.jp/api/v1/DeviceGroups << EOT { "devGrId": "90000001" } EOT</pre>
--	--

レスポンス

Code	説明
201	作成が成功し指定したデバイスグループIDが返る <pre>{ "devGrId": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/DeviceGroups

概要

作成済デバイスグループのID一覧を得る

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/DeviceGroups

レスポンス

Code	説明
200	登録済デバイスグループIDの一覧が返る <pre>{ "devGrIds": ["string", "string", ] }</pre>

400	パラメータにがであったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>
-----	--

GET /api/v1/DeviceGroups/{id}

概要

指定したIDのデバイスグループに登録されているデバイスIDの一覧を返す。URLの{id}部分にはデバイスグループIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/DeviceGroups/90000001

レスポンス

Code	説明
200	デバイスIDの一覧を返す <pre>{ "devIds": ["string", "string", ] }</pre>
400	パラメータに誤りがであったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/DeviceGroups/{id}

概要

登録済デバイスグループを削除する。URLの{id}部分にはデバイスグループIDを指定する。そのデバイスグループに登録されているデバイスがある場合はそれらも削除される。またチャンネルが設定済みであれば、そのチャンネルも削除される。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --request DELETE \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/DeviceGroups/90000001</pre>

レスポンス

Code	説明
200	正常処理のステータスが返る { "status": "Ok" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

Devices

POST /api/v1/Devices

概要

指定IDのデバイスを既存のデバイスグループに登録する。デバイスグループは予め作成してある必要がある。

【注意】1回のリクエストで複数のデバイスIDを指定できるが、デバイスの登録に時間がかかるため最大10デバイスまでとする。APIでは10を越えるデバイスの登録も受け付けるが、リクエスト処理に時間がかかるため中断する可能性がある。その場合はレスポンスが得られなくなる。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
リクエスト ボディ	登録済のデバイスグループと、登録するデバイスのIDを指定する { "devGrId": "string", "devIds": [

	<pre> "string", "string",] } </pre>
curlでの例	<pre> curl --header "Authorization: Bearer \${ADMIN_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Devices << EOT { "devGrId": "90000001", "devIds": ["00000001", "00000002"] } EOT </pre>

レスポンス

Code	説明
201	<p>デバイスが登録できた場合、接続に必要なIoT Hubのホスト名(hostname)、IoT HubでのデバイスID(hubId)、プライマリ接続文字列(key1)、セカンダリ接続文字列(key2)が返る</p> <pre> { "hostname": "string", "devGrId": "string", "devices": [{ "devId": "string", "hubId": "string", "key1": "string", "key2": "string" }, { "devId": "string", "hubId": "string", "key1": "string", "key2": "string" }, ] } </pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre> { "status": "Error", </pre>

	<pre>"message": "string" }</pre>
--	----------------------------------

GET /api/v1/Devices

概要

そのオーナーが登録済デバイスのID一覧を返す

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Devices

レスポンス

Code	説明
200	デバイスIDの一覧が返る <pre>{ "devId": ["string", "string", ] }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/Devices/{id}

概要

登録済デバイスの詳細を調べる。URLの{id}部分にはデバイスIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Devices/00000001

レスポンス

Code	説明
200	指定デバイスの、デバイスグループID、接続に必要なIoT Hubのホスト名(hostname)、IoT HubでのデバイスID(hubId)、プライマリ接続文字列(key1)、セカンダリ接続文字列(key2)が返る <pre>{ "devId": "string", "devGrId": "string", "hostname": "string", "hubId": "string", "key1": "string", "key2": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/Devices/{id}

概要

登録済デバイスを削除する。URLの{id}部分にはデバイスIDを指定する。もしそのデバイスが属するデバイスグループの最後の1個であった場合でデバイスグループにチャンネルが設定されている場合は、チャンネルも一緒に削除される。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --request DELETE \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Devices/00000001

レスポンス

Code	説明
200	正常処理のステータスが返る <pre>{ "status": "Ok" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

AppGroups

POST /api/v1/AppGroups

概要

指定したIDのアプリケーショングループを作成する

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
リクエスト ボディ	アプリケーショングループIDを指定する <pre>{ "devGrId": "a0000001" }</pre>
curlでの例	<pre>curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/AppGroups << EOT { "appGrId": "a0000001" } EOT</pre>

レスポンス

Code	説明
201	アプリケーショングループIDが返る <pre>{ "devGrId": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/AppGroups

概要

登録済アプリケーショングループIDの一覧を得る

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<code>curl --header "Authorization: Bearer \${OWNER_TOKEN}" \</code> <code>https://api.dexpf.jp/api/v1/AppGroups</code>

レスポンス

Code	説明
200	登録済アプリケーショングループIDの一覧が返る <pre>{ "appGrId": ["string", "string", ] }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/AppGroups/{id}

概要

指定したIDのアプリケーショングループに登録されているアプリケーションIDの一覧を返す。{id}の部分にはアプリケーショングループIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/AppGroups/a0000001</pre>

レスポンス

Code	説明
200	登録済アプリケーションIDの一覧を返す <pre>{ "appIds": ["string", "string", ] }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/AppGroups/{id}

概要

登録済アプリケーショングループを削除する。URLの{id}部分にはアプリケーショングループIDを指定する。そのアプリケーショングループに登録されているアプリケーションがある場合はそれらも削除される。またチャンネルが設定済みであれば、そのチャンネルも削除される。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --request DELETE \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/AppGroups/a0000001

レスポンス

Code	説明
200	正常処理のステータスが返る { "status": "Ok" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

Apps

POST /api/v1/Apps

概要

指定IDのアプリケーションを既存のアプリケーショングループに登録する。アプリケーショングループは予め作成済である必要がある。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
リクエスト ボディ	DEXPFからのメッセージを受信するアプリケーションの情報を指定する { "url": "string", "headers": ["string", "string"] }

curlでの例	<pre>curl --header "Authorization: Bearer \${ADMIN_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Apps << EOT { "appGrId": "string", "appId": "string", "url": "https://sample.example.jp/apps", "headers": ["Content-Type: application/json", "Authorization: Bearer 9he69Z6mrbiDaR22wrIZxIBzwQTTPh"] } EOT</pre>
---------	---

レスポンス

Code	説明
201	指定したアプリケーショングループIDと、アプリケーションIDが返る <pre>{ "appGrId": "string", "appId": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/Apps

概要

オーナーが登録したアプリケーションIDの一覧を得る

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Apps</pre>

レスポンス

Code	説明
200	アプリケーションIDの一覧が返る <pre>{ "appIds": ["string", "string", ] }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/Apps/{id}

概要

登録済アプリケーションの詳細を調べる。URLの{id}部分にはアプリケーションIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Apps/f0000001

レスポンス

Code	説明
200	アプリケーションの情報が返る <pre>{ "appGrId": "string", "appId": "string", "url": "string", "headers": ["string", "string", ] }</pre>

	}
400	<p>パラメータに誤りがあったことを示す。</p> <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/Apps/{id}

概要

登録済アプリケーションを削除する。URLの{id}部分にはアプリケーションIDを指定する。もしそのアプリケーションが属するアプリケーショングループの最後の1個であった場合にそのアプリケーショングループにチャンネルが設定されている場合は、チャンネルも一緒に削除される。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --request DELETE \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Apps/f0000001</pre>

レスポンス

Code	説明
200	<p>正常処理のステータスが返る</p> <pre>{ "status": "Ok" }</pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre>{ "status": "Error", "message": "string" }</pre>

Channels

POST /api/v1/Channels

概要

アプリケーショングループとデバイスグループの間にチャンネルを作成する。

- チャンネルの作成ではアプリケーショングループ側のオーナーとデバイスグループ側のオーナーを区別する必要がある。
- アプリケーション側、デバイス側のいずれでも実行できる。
- レスポンスでは、リクエスト側("appSide" または "devSide")が "enable" となり、他方が "disable" の状態となる。
- アプリケーション側、デバイス側とも同じオーナーが実行した場合は、両方とも "enable" となる。
- チャンネルを作成する場合は、アプリケーショングループに最低1個のアプリケーション、デバイスグループに最低1個のデバイスが登録されている必要がある。

パラメータ

Authorization ヘッダ	Bearer としてリクエストするオーナーアカウントのトークンを設定する
リクエスト ボディ	アプリケーション側とデバイス側を区別するために、それぞれのオーナーIDをappOwnIdとdevOwnIdとして設定する。 <pre>{ "appOwnId": "string", "appGrId": "string", "devOwnId": "string", "devGrId": "string" }</pre>
curlでの例	<pre>curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Channels << EOT { "appOwnId": "74724a65", "appGrId": "a0000001", "devOwnId": "e166479a", "devGrId": "90000001" } EOT</pre>

レスポンス

Code	説明
201	チャンネルの作成が成功した場合はその状態が返る <pre>{ "chanId": "string", "appOwnId": "string", "appGrId": "string", "devOwnId": "string", "devGrId": "string", "appSide": "string", "devSide": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

GET /api/v1/Channels

概要

登録済チャンネルIDの一覧を得る

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Channels

レスポンス

Code	説明
200	チャンネルIDの一覧が返る <pre>{ "chanIds": ["string", "string", ] }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{</pre>

	<pre>"status": "Error", "message": "string" }</pre>
--	---

GET /api/v1/Channels/{id}

概要

指定したIDのチャンネルの詳細を調べる。URLの{id}部分にはチャンネルIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Channels/abcdef01

レスポンス

Code	説明
200	チャンネルの詳細を返す <pre>{ "chanId": "string", "appOwnId": "string", "appGrId": "string", "devOwnId": "string", "devGrId": "string", "appSide": "string", "devSide": "string" }</pre>
400	パラメータに誤りがあったことを示す。 <pre>{ "status": "Error", "message": "string" }</pre>

PUT /api/v1/Channels/{id}

概要

チャンネルを作成したオーナーの相手側のオーナー(例：アプリケーション側が作成した場合はデバイス側)がそのチャンネルでの通信を許可する。URLの{id}部分にはチャンネルIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --request PUT \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Channels < /dev/null</pre>

レスポンス

Code	説明
201	<p>チャンネルが通信できるようになった状態を示す。appSide と devSide は両方とも enable となる。</p> <pre>{ "chanId": "string", "appOwnId": "string", "appGrId": "string", "devOwnId": "string", "devGrId": "string", "appSide": "enable", "devSide": "enable" }</pre>
400	<p>パラメータに誤りがあったことを示す。</p> <pre>{ "status": "Error", "message": "string" }</pre>

DELETE /api/v1/Channels/{id}

概要

指定したチャンネルIDのチャンネルを削除する。URLの{id}部分にはチャンネルIDを指定する。

パラメータ

Authorization ヘッダ	Bearer としてオーナーアカウントのトークンを設定する
curlでの例	<pre>curl --request DELETE \ --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Channels/abcdef01</pre>

レスポンス

Code	説明
200	正常終了 { "status": "Ok" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

Version

GET /api/v1/Version

概要

DEXPFのバージョンを調べる

パラメータ

curlでの例	curl --header "Authorization: Bearer \${OWNER_TOKEN}" \ https://api.dexpf.jp/api/v1/Version
---------	--

レスポンス

Code	説明
200	DEXPFのシステムIDとバージョンが返る { "systemId": "string", "version": "string" }

Admin

POST /api/v1/Admin/Setup

概要

Adminアカウントを作成するとともに、DEXPFの初期設定を行う。

【注意】このリクエストは1回限り実行できる。2回目以降の実行はエラーとなる。

パラメータ

リクエスト ボディ	Adminのユーザー名とパスワードを指定する { "username": "string", "password": "string" }
curlでの例	curl --header "Content-Type: application/json" \ --data @- \ https://api.dexpf.jp/api/v1/Admin/Setup << EOT { "username": "admin@example.jp", "password": "JYLzcaKmdF5GuQ429XJ3YZ3HxcTwHp" } EOT

レスポンス

Code	説明
201	AdminアカウントのオーナーIDとDEXPFのシステムIDが返る { "systemId": "string", "ownerId": "string" }
400	パラメータに誤りがあったことを示す。 { "status": "Error", "message": "string" }

改訂履歴

バージョン	日付	内容
Ver 1.1-05	2019-12-20	初版
Ver 1.1-06	2020-01-10	改訂履歴を追加 DEXPF API 全般部分のエラーに関する記述を修正 DEXPF API のレスポンスヘッダに関する説明を追加 トークンの扱いに関する注意を追加 例の中でIDが7桁になっているものがあつたため修正
Ver 1.1-07	2020-03-27	正式版としてリリース
Ver 1.1-08	2020-06-29	DEXPFユーザーズガイドの改訂に合わせたバージョン更新