BraveGATE CORE ユーザーズガイド

株式会社Braveridge

バージョン2023/08/07

目次

1. 概要	1
1.1. 本ドキュメントについて	1
1.2. 製品概要	1
1.3. API機能概要	1
2. 凡例	2
2.1. shellの実行表記について	2
2.2. 文字列の例表記	3
3. 機能リファレンス	4
3.1. API仕様書について	4
3.2. API使用方法	4
3.2.1. 基本URLについて	4
3.2.2. 通信仕様について	4
3.2.3. レスポンスの表示について(ページネーション)	5
3.3. Auth APIについて	5
3.3.1. ルートアカウントによるTokenの発行 (POST /auth).	5
3.3.2. authkey認証情報 APIアクセス用Tokenの発行(POST /auth)	6
3.4. グループ機能	8
3.4.1. グループとは	8
3.4.2. グループの登録(POST /groups)	8
3.4.3. group IDの取得(GET /groups)	9
3.4.4. グループ個別情報の取得(GET /groups/{group_id})	10
3.4.5. グループ情報更新(PUT /groups/{group_id})	11
3.4.6. グループ削除(DELETE /groups/{group_id})	11
3.5. ルーター管理	13
3.5.1. ルーターの登録(POST /routers)	13
3.5.2. ルーター情報一覧(GET /routers)	
3.5.3. ルーター詳細取得(GET /routers/{router_id})	
3.5.4. ルーターの情報更新(PUT /routers)	
3.5.5. ルーターの削除(DELETE /routers/{router_id})	
3.6. デバイス管理	
3.6.1. デバイスの登録(POST /devices)	
3.6.2. デバイス情報一覧(GET /devices)	
3.6.3. デバイス詳細取得(GET /devices/{device_id})	
3.6.4. デバイスの情報更新(PUT /devices/{device_id})	
3.6.5. デバイス削除(DELETE /devices/{device_id})	
3.7. アプリケーション管理(アップリンクデータ転送)	
3.7.1. アプリケーション(Webhook機能の受信先)の設定 (POST /applications)	
3.7.2. Webhookの形式(HTTP ヘッダー)	

	2.7.2 Mabbaakの形式(ニーク形式 ITERIE ごじょうについて)	20
	3.7.4. Webhookの形式(テータ形式、ELTRESテバイスについて)	31
	3.7.5. アプリケーション(データ受信先のAzure Blob Storage)の設定 (POST /applications)	32
	3.7.6. Azure Blob Storageへ保存するデータ形式	. 35
	3.7.7. アプリケーションの情報一覧 (GET /applications)	. 35
	3.7.8. アプリケーション詳細取得 (GET /applications/{application_id})	36
	3.7.9. アプリケーションの情報更新(PUT /applications/{application_id})	37
	3.7.10. アプリケーション削除(DELETE /applications/{application_id})	. 38
3	.8. コマンド送信 (Downlink)	. 39
	3.8.1. コマンド送信(POST /commands)	. 40
	3.8.2. コマンド結果一覧の取得(GET /commands)	. 40
	3.8.3. コマンド結果の詳細の取得(GET /commands/{command_id})	. 43
	3.8.4. コマンド実行時のエラーケースについて	. 47
3	.9. モバイル通信量の取得 (BraveSIM)	. 49
	3.9.1. IMSIの確認方法	. 49
	3.9.2. モバイル通信量の取得 (GET /sims)	. 49
	3.9.3. router id を利用したモバイル通信量の取得 (GET /routers/{router_id}/data_usage)	. 51
4. ち	ダウンリンクコマンド凡例	. 54
4	1. センサデータの即時取得	. 54
4	.2. センサデータのUplink間隔設定	. 55
4	3. データダウンロード(デバイスのみ)	. 56
4	4. デバイスフィルター	57
	4.4.1. Scan対象フィルタ上書き設定(DEVICE ID)(ルーターのみ)	. 57
	4.4.2. Scan対象フィルタ設定追加(DEVICE ID) (ルーターのみ)	. 58
	4.4.3. Scan対象フィルタ設定クリア(DEVICE ID) (ルーターのみ)	. 59
4	5. 電源オペレーション	. 60
	4.5.1. 電源オフ	. 60
	4.5.2. ルーター再起動(ルーター)	. 61

1. 概要

1.1. 本ドキュメントについて

株式会社Braveridge (以下「当社」と表記) が提供するBraveGATE CORE のユーザーズガイドです。

1.2. 製品概要

BraveGATE CORE は、ダイナミックなデバイス制御をRESTful APIにより提供するシステムです。以下の 様な機能を提供します。

デバイス制御APIの機能(一部抜粋)

- 即時Uplink要求
- Uplink間隔設定
- Uplink期間設定
- Uplink閾値設定
- Uplink有効化 · 無効化
- センサー有効化・無効化
- 測定レンジ設定
- レジスタ設定
- ファームウェアアップデート
- 再起動

ルーター制御APIの機能(一部抜粋)

- デバイスフィルタリング
- ファームウェアアップデート
- 再起動

1.3. API機能概要

- HTTPSによるセキュアな通信
- ・閉域網接続によるセキュアなIoTシステムの構築 デバイス、ルーターを閉域網に接続しBraveGATEサービス経由でのみアクセス可能とすることにより、セキュアなIoTシステムの構築できます。
- JSON形式によるデータの受け渡し JSON形式によるデータの受け渡しにより、さまざまなシステムとの連携を容易にします。
- Webhook登録によるデータの受け渡し
 Webhookを登録することにより、デバイスのセンサー情報を定期的に自システムに提供することが可能です。

2. 凡例

2.1. shellの実行表記について

本ユーザーガイドでは、UNIX/Linux上でのsh(Bourne Shell)上でのcurlコマンド実行例を表記します。 Mac OS Xのターミナルでも同様に実行可能です。

Windows 10 Ver.1803以降については、コマンドプロンプトより「curl.exe」にて同様に実行可能です。 読み替えて使用してください。

先頭のshell promptは、一般的な例示として" \$ "にて開始します。

例) ○○を実施する。

\$ curl https://api.example.com

結果表示例があるものについては、以下の様にコマンド実行後の画面表示を続けて記載しますが、JSONの 表示については可読性を考慮しファイル形式として記載します。

例)

```
$ curl https://api.example.com/v1/foo
{
    "status": "202"
}
```

表示例中 -d @file.jsonの様に、JSONファイルを指定する場合はコマンド事例の後に、ファイルの内容を 続けて記載します。

\$ curl https://api.example.com/v1/foo -d @foo.json

ファイル内容: foo.json

```
{
    "authkey_id" : "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgh",
    "authkey_secret" :
"ABCdE0f1ghi2j3KLMNoPQrStuV45wxYZ6aBcDEfGHIJk8lmN0pqr9S0Tuv1wXyzabC",
    "timeout_seconds" : 86400
}
```

表示例中、正常·異常動作があるものについては、以下の様にコマンド実行後の画面表示を場合分けのコメントと共に続けて記載します。

\$ curl https://api.example.com/v1/bar

成功した場合何も表示されない

失敗した場合、以下の様な表示となる。

```
{
    "errors": [
        "apikey is invalid."
    ]
}
```

2.2. 文字列の例表記

RFC3092 に従い、意味のない文字列として foo bar baz qux ... を使用します。

http://www.rfc-editor.org/rfc/rfc3092.txt

3. 機能リファレンス

3.1. API仕様書について

APIのREST仕様については以下URLにて公開されておりますので、ご参照ください。

http://developer.braveridge.io

各リクエストURLに記載されている、Parameters 、Responses の Description 部に Exmaple Value として 記述例が、Model としてJSONの仕様が記載されておりますでの、ご参照下さい。

3.2. API使用方法

3.2.1. 基本URL について

API提供の基本URLについては

scheme: https
host: api.braveridge.io
path: /v1/[command]

https://api.braveridge.io/v1

としてAPIを提供します。 注)「/v1」とAPIのバージョン番号がpathの先頭に付与されて提供されています。

3.2.2. 通信仕様について

API通信はHTTPS経由で行われます。リクエストとレスポンスは両方ともJSONとしてフォーマットされ、両方のコンテンツタイプは application/jsonです。

パラメータ通信に用いる文字コードはUTF-8とします。リクエストパラメータおよびレスポンスパラメー タはJSON.[ECMA-404 The JSON Data InterchangeStandard]とし、そのルート要素はrequestおよ びresponseとします。HTTP要求ヘッダーAcceptは、"application/json" もしくは、"*/*"を指定して下さい。

例) リクエスト時のHTTP Header

3.2.3. レスポンスの表示について(ページネーション)

一覧表示など複数の情報を表示する場合、RESTful API のページネーションを元にページングを行います。

例) ページネションの表示

```
...
    "total": 29, 1
    "pages": 3, 2
    "limit": 10, 3
    "current_page": 1, 4
    "next_page": "https://api.braveridge.io/v1/routers?limit=10&page=2" (5)
}
```

1 トータルで何件の配列数

2 全体のページ数

3 1ページに表示する最大数(defaultで10件)

- ④ 現在表示しているページ
- ⑤ 次のページのURL

デフォルトの表示件数は10件となります。 APIのリクエスト時にquery stringにて、limitの件数、page数 を指定することができます。 limitの件数は、1-1000の間で指定することが可能です。

例)件数1000を指定した場合のURL

https://api.braveridge.io/v1/routers?limit=1000

例) 100件表示の3ページ目を表示した場合のURL

https://api.braveridge.io/v1/routers?limit=100&page=3

3.3. Auth APIについて

BraveGATEのAPIを使用するために、ルートアカウント情報 (email_address, password) もしくは、ルートアカウントに紐付くauthkey認証情報 (authkey_id, authkey_secret) を使用してAPIにアクセスするTokenを発行する Auth APIについて説明します。

NOTE

ルートアカウント情報、authkey認証情報について、サービス申込時に当社より発行され ます。authkey認証情報については、提供予定のBraveGATE ユーザーポータルより、複 数発行し認証情報毎の管理を行う機能を提供する予定です。

3.3.1. ルートアカウントによるTokenの発行 (POST /auth)

発行されたユーザー名(メールアドレス)と初期パスワードを、ルートアカウント認証情報と言います。 ルートアカウント認証情報を使用し、 POST /auth することにより操作に必要なAPIのトークン認証用の

apikey と token を発行します。

```
実行コマンド
```

ファイル内容: auth_first.json

```
{
    "email_address" : "foo@example.com",
    "password" : "example_password",
    "timeout_seconds" : 86400
}
```

```
NOTE ルートアカウント認証情報を使って発行した apikey と token を使ってすべての操作を行
えますが、顧客ポータルなどBraveGATEのすべてのリソースへの完全かつ無制限なアク
セスが可能になるため、基本的にAPIアクセスにルートアカウント認証情報を使わないよ
うにして下さい。authkey_id と authkey_secret は、BraveGATE ユーザーポータル(提供
予定)より複数発行できますので、アプリケーションサーバ毎に使い分ける等の使い方が
できます。
```

3.3.2. authkey認証情報 APIアクセス用Tokenの発行(POST /auth)

発行された authkey_id と authkey_secret で認証を行い、apikey と token を POST /auth にて取得します。

以下の、authkey_id 、 authkey_secret が発行されていた場合

authkey_id: AICDEFGHIJKLMNOPQRSTUVWXYZabcdefgh authkey_secret: ASCdE0f1ghi2j3KLMNoPQrStuV45wxYZ6aBcDEfGHIJk81mN0pqr9S0Tuv1wXyzabC

この認証情報を使用し、以下の様にコマンドを実行します。

timeout_seconds を指定して、24時間(86,400秒)の有効期限を設定しています。 timeout_secondsを指定 することにより、指定時間内であればAPIアクセスする度に apikey,token を毎回発行する必要はありません。

```
実行コマンド
```

ファイル内容: auth.json

```
{
    "authkey_id" : "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefgh",
    "authkey_secret" :
"ABCdE0f1ghi2j3KLMNoPQrStuV45wxYZ6aBcDEfGHIJk8lmNOpqr9S0Tuv1wXyzabC",
    "timeout_seconds" : 86400
}
```

各APIの使用には取得した apikey をHTTP Headerの X-Braveridge-API-Key に設定し、 token をHTTP Headerの X-Braveridge-Token に設定して以下の様に使用します。 apikey は、AK を接頭語として始る文字列、token は、TK を接頭語として始る文字列となります。

例) key,tokenを設定して実行コマンド

3.4. グループ機能

3.4.1. グループとは

複数のルーターとデバイスをグループでまとめて管理できる機能です。

く企業A>が【扱	処点1]と[拠点2] にルータとデバイスを	を設置した場合
Account (企業A)			
Group(=拠点1)	Group(=拠点2	2)
Router	Device	Router	Device
Router	Device	Router	Device
Router	Device	Router	Device
Router	Device	Router	Device
Router	Device	Router	Device

グループに対して、次の様な操作が可能です。

- グループ毎にアプリケーションを登録
- グループを指定して、同一コマンド実行

3.4.2. グループの登録(POST /groups)

新規のグループを POST /groups にて作成します。

例) グループを作成

Braveridge

ファイル内容: create_group.json

```
{
    "name": "floor-1 group",
    "is_default": false ①
}
```

① デフォルトグループ(group_id 指定省略時に選択されるグループ)の選択、booleanで指定します。

3.4.3. group IDの取得(GET /groups)

登録されているグループの一覧を GET /groups にて取得します。

例) グループー覧を取得

```
$ curl -s https://api.braveridge.io/v1/groups -X GET -H X-Braveridge-API-Key:
<RESPONSE>
{
  "groups": [
     {
        "name": "",
        "is_default": false,
        "created_at": "2019-10-02T14:14:16+09:00",
        "updated at": "2019-10-02T14:14:16+09:00"
     },
     {
        "name": "floor-1 group",
        "is_default": false,
        "created at": "2019-10-02T14:09:57+09:00",
        "updated_at": "2019-10-02T14:09:57+09:00"
     },
     {
        "name": "",
        "is default": true,
        "created_at": "2019-06-14T16:08:51+09:00",
        "updated_at": "2019-06-14T16:08:51+09:00"
     }
  ],
  "total": 3,
  "pages": 1,
   "limit": 10,
  "current_page": 1,
  "next_page": null
}
```

is_default が true のグループがデフォルトグループとなります。 デフォルトグループは、group_idを指 定しない場合に選択されます。

3.4.4. グループ個別情報の取得(GET /groups/{group_id})

登録されているグループ個別の情報を GET /groups/{group_id} にて取得します。

例) グループ情報を取得

3.4.5. グループ情報更新(PUT /groups/{group_id})

登録されているグループの名前を変更したり、デフォルトグループを PUT /groups/{group_id} にて変更 します。

例) グループの名前を変更、デフォルトグループを指定して更新

ファイル内容: change_group.json

```
{
    "name": "floor-2 group",
    "is_default": true
}
```

3.4.6. グループ削除(DELETE /groups/{group_id})

登録されているグループを、DELETE /groups/{group_id} にてBraveGATEより削除します。 ルーター、デ バイスが一つでも所属しているグループは削除できません。グループを削除前に、所属しているルータ ー、デバイスのグループを変更して下さい。 例) グループを削除

3.5. ルーター管理

3.5.1. ルーターの登録(**POST** /**routers**)

ルーターを使用するためには、Braveridgeより発行された router_id と、registration_code を使用して POST /routers にてBraveGATEへ登録する必要があります。

```
以下の、router_id、 registration_code が発行されていた場合
```

```
router_id: bb054321
registration_code: abcd1234
```

```
例) ルーターを登録
```

ファイル内容: router.json

```
{
    "router_id": "bb054321",
    "registration_code": "abc1234",
    "group_id": "",
    "name": "foo bar"
}
```

group_idを指定しなかった場合、デフォルトグループに指定されたグループに所属します。

3.5.2. ルーター情報一覧(GET /routers)

BraveGATEに登録されたルーターの一覧を GET /routers にて取得します。

例) ルーターの情報一覧を取得

```
$ curl -s https://api.braveridge.io/v1/routers -X GET -H X-Braveridge-API-Key:
<RESPONSE>
{
   "routers": [
      {
         "router_id": "bb054321",
         "name": "foo",
         "imsi": "440012345678901",
         "rssi": 0, // LTEの電波強度を表します。
         "battery": -1, // バッテリー残量(%)を表します。①
         "sim_status": "ready", // SIMカードの状態。①
         "segment_id": 0, // 回線、地域等の接続区分を表します。
         "created_at": "2019-03-25T17:24:20+09:00",
         "updated_at": "2019-03-25T17:24:20+09:00"
      },
      {
         "router_id": "bb054322",
         "name": "bar",
         "group_id": "GRqvnp8qkYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
         "imsi": "440012345678902",
         "rssi": -69,
         "battery": 48,
         "sim_status": "active",//
SIMカードの状態。ルーター登録後一度でも通信を行なうと、"active"となります。
         "segment_id": 0,
         "created at": "2019-03-25T15:27:29+09:00",
         "updated_at": "2019-03-25T15:27:29+09:00"
      },
   ],
   "total": 2,
   "pages": 1,
   "limit": 10,
   "current_page": 1,
   "next_page": null
}
```

 ルーター登録後いちども通信していない場合は、battery'は、-1`を示します。また sim_status' は、'ready'となります。

Query パラメーター

group id をQuery パラメーターとして指定する事により、指定されたgroup idに所属するrouter の一覧 を取得する事ができます。 例) group_id を指定して、ルーターの情報一覧を取得

```
$ curl -s
<RESPONSE>
{
  "routers": [
    {
      "router_id": "bb054321",
      "name": "foo",
      "imsi": "440012345678901",
      "rssi": 0,
      "battery": -1,
      "sim_status": "ready",
      "segment_id": 0,
      "created_at": "2019-03-25T17:24:20+09:00",
      "updated_at": "2019-03-25T17:24:20+09:00"
    }
  ],
  "total": 1,
  "pages": 1,
  "limit": 10,
  "current_page": 1,
  "next_page": null
}
```

3.5.3. ルーター詳細取得(GET /routers/{router_id})

BraveGATEに登録されたルーター個別の情報を GET /routers/{router_id} にて取得します。

例) ルーターの情報詳細の取得

```
$ curl -s https://api.braveridge.io/v1/routers/bb054321 -X GET -H X-Braveridge-API-
<RESPONSE>
{
  "router_id": "bb054321",
  "name": "",
  "imsi": "440012345678901",
  "rssi": -68,
  "battery": 30,
  "sim_status": "ready",
  "segment_id": 0,
  "created_at": "2019-03-25T17:24:20+09:00",
  "updated at": "2019-03-25T17:24:20+09:00"
}
```

3.5.4. ルーターの情報更新(PUT /routers)

ルーターの名前や所属させるグループを変更する場合には、 PUT /routers/{router_id} にて情報の更新 を行います。

例)ルーターの所属するグループ、名前を変更

ファイル内容: router_put.json

3.5.5. ルーターの削除(DELETE /routers/{router_id})

ルーターの登録を、DELETE /routers/{router_id} にてBraveGATEから削除します。また、ルーターを削除することにより、グループやアカウントへの紐付けの情報が削除されます。

例)ルーターの登録を削除

3.6. デバイス管理

3.6.1. デバイスの登録(POST /devices)

デバイスを使用するためには、予め発行された device_id と、registration_code を使用してBraveGATE へ登録する必要があります。

```
以下の、device_id、 registration_code が発行されていた場合
```

```
device_id: 2468801234012345
registration_code: abcd1234
```

例) デバイスを登録

ファイル内容: device.json

```
{
    "device_id":"2468801234012345",
    "registration_code":" abcd1234",
    "group_id" : "",
    "name": "foo bar"
}
```

group_idを指定しなかった場合、デフォルトグループに指定されたグループに所属します。

3.6.2. デバイス情報一覧(GET /devices)

登録されているデバイスの一覧を、GET /devices にて取得します。デバイスの情報として実行可能なコマンドの一覧を表示します。

例) デバイスの情報一覧を取得

```
{
                "sensor_id": "0004",
                "name": "Temperature sensor",
                "executable_commands": [ ①
                    "SEND_DATA_AT_ONCE",
                    "SET_UPLINK_INTERVAL",
                    "SET_UPLINK_TERM",
                    "SET_UPLINK_THRESHOLD"
                ]
            }
        ],
        "created at": "2018-11-29T14:49:04+09:00",
        "updated_at": "2018-11-29T14:49:04+09:00"
   },
    {
        "device_id": "2468804321543210",
        "name": "",
        "group_id": "GRqvnp8qkYYYYYYYYYYYYYYYYYYYYYYYYYYYY,
        "sensors": [
            {
                "sensor_id": "0005",
                "name": "Humidity sensor",
                "executable_commands": [ ①
                    "SEND_DATA_AT_ONCE",
                    "SET_UPLINK_INTERVAL",
                    "SET_UPLINK_TERM",
                    "SET UPLINK THRESHOLD"
                ]
            }
        ],
        "created_at": "2018-11-29T14:49:04+09:00",
        "updated_at": "2018-11-29T14:49:04+09:00"
   },
],
"total": 2,
"pages": 1,
"limit": 10,
"current_page": 1,
"next_page": null
```

① デバイスに対する実行可能なコマンド一覧

}

Query パラメーター

group id をQuery パラメーターとして指定する事により、指定されたgroup idに所属するdevice の一覧 を取得する事ができます。

例) group_id を指定して、デバイスの情報一覧を取得

```
$ curl -s
<RESPONSE>
{
  "devices": [
     {
       "device_id": "2468801234012345",
       "name": "",
       "sensors": [
         {
            "sensor id": "0004",
            "name": "Temperature sensor",
            "executable_commands": [
               "SEND_DATA_AT_ONCE",
               "SET_UPLINK_INTERVAL",
               "SET_UPLINK_TERM",
               "SET_UPLINK_THRESHOLD"
            ]
         }
       ],
       "created_at": "2018-11-29T14:49:04+09:00",
       "updated at": "2018-11-29T14:49:04+09:00"
     }
  ],
  "total": 1,
  "pages": 1,
  "limit": 10,
  "current_page": 1,
  "next_page": null
}
```

LPWA(ELTRES) 対応デバイスについて

LPWA(ELTERS) 対応にデバイスについては、specification key の情報が付加されます。

```
"specification": { /* LPWA通信個別の情報を付加します。*/
    "lpwa_type": "<LPWAの種類(ELTRES)",
    "lfourId": <端末固有の識別ID(最大40bitの整数)>,
    "fw_version": "<デバイスのFirmwareバージョン. x.x.x>"
}
```

例) デバイスの情報一覧を取得(ELTRESデバイスが登録されている場合)

以下に、ELTRESのデバイスが一覧に含まれる場合の表記例を示します。

```
$ curl -s https://api.braveridge.io/v1/devices -X GET -H X-Braveridge-API-Key:
<RESPONSE>
{
   "devices": [
      {
         "device_id": "2468801234012345",
         "name": "",
         "sensors": [
            {
                "sensor id": "0004",
                "name": "Temperature sensor",
                "executable_commands": [] ①
            }
         ],
         "specification": {
             "lpwa type": "ELTRES",
            "lfourId": 12374567890,
            "fw version": "1.2.3"
         },
         "created_at": "2018-11-29T14:49:04+09:00",
         "updated_at": "2018-11-29T14:49:04+09:00"
      },
      {
         "device_id": "2468804321543210",
         "name": "",
         "group_id": "GRqvnp8qkYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
         "sensors": [
            {
                "sensor_id": "0005",
                "name": "Humidity sensor",
                "executable commands": [
```

```
"SEND_DATA_AT_ONCE",
                         "SET_UPLINK_INTERVAL",
                         "SET_UPLINK_TERM",
                         "SET_UPLINK_THRESHOLD"
                    ]
                }
            ],
            "created_at": "2018-11-29T14:49:04+09:00",
            "updated_at": "2018-11-29T14:49:04+09:00"
        },
    ],
    "total": 2,
    "pages": 1,
    "limit": 10,
    "current_page": 1,
    "next_page": null
}
```

① ELTRESはdownlinkをサポートしないため実行できるコマンドはなし

3.6.3. デバイス詳細取得(GET /devices/{device_id})

登録されているデバイスの詳細情報を取得します。 BraveGATEに登録されたデバイス個別の情報を GET /devices/{device_id} にて取得します。

例) デバイスの詳細情報の取得

```
$ curl -s https://api.braveridge.io/v1/devices/2468801234012345 -X GET -H X-
<RESPONSE>
{
   "device id": "2468801234012345",
   "name": "Tracker09",
   "group_id": "GRec3b8f31f2caef3a749353685afa7555",
   "sensors": [
      {
         "sensor_id": "000b",
         "name": "GPS",
         "executable commands": [
            "SEND_DATA_AT_ONCE",
            "SET UPLINK INTERVAL"
         1
      },
      {
         "sensor_id": "001e",
         "name": "Full Color LED",
         "executable commands": [
            "SET SENSOR RANGE"
         ]
      }
   1,
   "created_at": "2019-02-21T14:16:31+09:00",
   "updated_at": "2019-02-21T14:16:31+09:00"
}
```

LPWA(ELTRES)対応デバイスについては、デバイス情報一覧(GET /devices)と同様に`specification` key の情報が付加されます。詳細は、 LPWA(ELTRES) 対応デバイスについて を参照下さい。

3.6.4. デバイスの情報更新(PUT /devices/{device_id})

デバイスの名前や所属させるグループを変更する場合には、 PUT /devices/{device_id} にて情報の更新 を行います。 例)デバイスの所属するグループ、名前を変更

ファイル内容: device_put.json

3.6.5. デバイス削除(DELETE /devices/{device_id})

デバイスの登録をBraveGATEから DELETE /devices/{device_id} にて削除します。 デバイスを削除する ことにより、グループやアカウントへの紐付けなどすべての情報が削除されます。

3.7. アプリケーション管理(アップリンクデータ転送)

デバイスのセンサーデータは、あらかじめ指定したユーザーアプリケーションのURLにBraveGATE CORE からJSONデータがPOSTされることによりセンサーデーター取得することができるwebhook機能と、デー タを受け取るazure blobストレージを指定することができるazure blob機能が用意されています。 デバイ スに設定されたuplink間隔による定期的な取得や、閾値設定による取得、即時uplink命令を使った即時取 得などにてデータの取得が可能です。

3.7.1. アプリケーション(Webhook機能の受信先)の設定 (POST /applications)

webhookを受け取るアプリケーションのURLを、POST /v1/applications にて設定します。webhookに 指定するURLは、group id毎に指定できます。

 登録したアプリケーションのIDをappication_idとして返します。このIDを利用して、詳細情報の取 得、更新、削除に使用します。

ファイル内容: webhook.json

① httpも指定可能ですが、Production環境で使用することは推奨しません。

② 任意の文字列。token に指定できる文字は、token68(RFC 7235)に準拠します。

③ group_idは、optional。指定しなければ、default group となります。

urlには、query string等のURL表記として許容される文字列を含む事ができます。例) https://example.com/hook/aaabbbccc?key=foovar

TIP tokenを利用できない場合になんらかの認証情報を付加したい場合や、アプリケーションの 識別を行いたい場合等に使用できます。

API トークンの指定について

token を指定することにより、アプリケーションのURLに対してAPIトークンのヘッダーを付与し たwebhook を受信する事ができるようになります。

以下のヘッダーが付加されます。

X-Braveridge-Webhook-Token

ユーザー独自のヘッダーを指定することはできません。

定期的に token を変更する等の運用を行う場合は、変更後のwebhookから反映さ IMPORTANT れるため、受信するシステムにて新旧両方のトークンを認識できるような設計を する事を推奨します。

3.7.2. Webhookの形式(HTTP ヘッダー)

主な HTTP のヘッダーは、以下の様になります。

POST / HTTP/1.1 Host: foo.example.com Content-Length: 123 X-Braveridge-Webhook-Token: ① Accept: application/json, text/plain, */* Content-Type: application/json;charset=utf-8 User-Agent: BraveGATE-webhook/1.0.0

① アプリケーション登録時に token パラーメータを指定した場合は、このヘッダーが付与されます。

BraveGATE COREとwebhook送信先のapplicationとの通信要件については、RESTに従ったアーキテクチャとなります。

3.7.3. Webhookの形式(データ形式、LTE/BLE デバイスについて)

webhookにて受け渡されるJSONの定義

デバイスからBraveGATE CORE経由でwebhook機能(POST)にて渡されるデータはJSON形式であり、下記 の仕様となります。

```
{
 "group": {
   "group_id": "<登録デバイスが所属しているgourp_id>",
   "name": "<グループ名>"
 },
 "application": {
   "application_id": "<登録されたアプリケーションのapplictaion_id",
   "name": "<指定したアプリケーション名"
 },
 "router": {
   "router_id": "<デバイスからアップリンクを受信し BraveGATE CORE
に送信したrouter id>"
   "imsi": "IMSI 15桁の番号",
   "rssi": <LTEの電波強度 0 から-120>,
   "battery": <バッテリ残量(%)>
 },
 "device": {
   "device_id": "センサーデータを送信したdevice_id",
   "sensor_id": "センサー番号",
   "sensor_name": "センサー名",
   "rssi": デバイスとルーター間のBluetooth電波強度,
   "data": { /* 各センサーにより内容が異なります。*/
      "センサーを表す名前のkey": センサー値
      . . . .
   }
 },
 "uplink_id": "アップリンク固有ID(UUID形式)",
 "date": "アップリンクされてきた日時(ISO 8601)"
}
```

例)温度センサーのJSONデータ

例として温度センサーのデータ形式を上げます。基本的には

- .device.sensor_id
- .device.sensor_name
- .device.data

のキーの内容が、各デバイスのセンサーの種類によって異なります。 詳しくは、各デバイスのAPIコマン ド仕様書を参照下さい。

```
{
   "group": {
      "name": "Test Group"
   },
   "application": {
      "name": "Test Webhook"
   },
   "router": {
      "router_id": "bb000001",
      "imsi": "440000000000001",
      "rssi": -50,
      "battery": 75
   },
   "device": {
      "device_id": "2468800004000004",
      "sensor id": "0004",
      "sensor_name": "Temperature sensor",
      "rssi": -30,
      "data": {
         "temperature": 12.34
      }
   },
   "uplink_id": "1111aaaa-4222-bbbb-3333-cccc4444dddd",
   "date": "2018-10-23T00:00:00+09:00"
}
```

webhookによる大きいサイズのデータの受け渡しについて

ー定のサイズを越えるセンサーデータを送信するデバイスからの受け渡しについては、有効期限付きURL を発行して別途HTTPにてダウンロードするURL渡しの方式にてセンサーデンターの受け渡しを行ないま す。URL受け渡しを適用するセンサーデータの大きさについては、個々のデバイスの仕様によって異なり ます。(概ね240byte以上のデータ) 詳細は各デバイスのAPI仕様書を参照下さい。

```
{
 "group": {
   "group_id": "<登録デバイスが所属しているgourp_id>",
   "name": "<グループ名>"
 },
 "application": {
   "application_id": "<登録されたアプリケーションのapplictaion_id",
   "name": "<指定したアプリケーション名"
 },
 "router": {
   "router id": "<デバイスからアップリンクを受信し BraveGATE CORE
に送信したrouter id>"
   "imsi": "IMSI 15桁の番号",
   "rssi": <LTEの電波強度 0 から-120>,
   "battery": <バッテリ残量(%)>
 },
 "device": {
   "device_id": "センサーデータを送信したdevice_id",
   "sensor_id": "センサー番号",
   "sensor_name": "センサー名",
   "rssi": デバイスとルーター間のBluetooth電波強度,
   "data": { /* 有効期限付きURLが渡されます。*/ ①
      "url": "https://assets.braveridge.io/images/<アップリンク固有ID(UUID形式
)>/<ファイル名.(jpg|bin)>?Expires=<有効期限(unixtime)>&.....認証key等 guery
string..." (2)
      "contentType": "<(image/jpg|application/octet-stream)>", 3
      "contentLength":
   }
 },
 "uplink_id": "アップリンク固有ID(UUID形式)",
 "date": "アップリンクされてきた日時(ISO 8601)"
}
```

- ① URLの有効期限は、webhookの送信時間より5分となります。webhookを受信後、5分以内にデータ を取得して下さい。
- ② ファイル名の拡張子については、データの種類により異なります。詳細は各デバイスのAPI仕様書を参照下さい。
- ③ contentType については、データの種類により異なるmime type が設定されます。詳細は各デバイスのAPI仕様書を参照下さい。
- 例)カメラセンサーのJSONデータ
- 例として温度センサーのデータ形式を上げます。基本的には
 - .device.sensor_id
 - .device.sensor_name
 - .device.data

BraveGATE CORE ユーザーズガイド

```
Braveridge
```

のキーの内容が、各デバイスのセンサーの種類によって異なります。 詳しくは、各デバイスのAPIコマンド仕様書を参照下さい。

```
"aroup": {
       "name": "Test Group"
   },
    "application": {
       "name": "Test Webhook"
   },
   "router": {
       "router_id": "bb000001",
       "imsi": "44000000000001",
       "rssi": -50,
       "battery": 75
   },
   "device": {
       "device_id": "2468801234567890",
       "sensor_id": "0051",
       "sensor name": "Still Camera",
       "rssi": -30,
       "data": {
           "url": "https://assets.braveridge.io/images/3140fd4a-d21b-4705-a081-
81e1a859f1e6/image.jpg?Expires=1661510824&Policy=eyJTdGF0ZW1lbnQi0lt7IlJlc291cmNlIjoia
HR0cHM6Ly9hc3NldHMuYnJhdmVyaWRnZS5pby9pbWFnZXMvMzE0MGZkNGEtZDIxYi00NzA1LWEw0DEt0DF1MWE
4NTlmMWU2L2ltYWdlLmJpbiIsIkNvbmRpdGlvbiI6eyJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6M
TY2MTUxMDgyNH19fV19&Signature=haBHSUFNzc9evcvWEBWPFUNtRJ0tPvW8tLkDIubYrCMq6ClUj14y0QN2
YNwaZf~9xMhq7TZ2MVq~NwLEHLRWKYFX~1dPOWs05fJh9L~H8Mhpcox9RzeteUreXq2iqwWS0ojgRaMiN3H6oV
2eHVV9YVKnV~jkLWcrr6yNpNWSQKNYA96t2XCUfPiYKOdnwUd6yJr~qzBlKn2MRIG-
ZoedJagnszO30MH4Jkr9VkgrZor79re8IE~WdQLapEgh55BN9gMd1VUoZXct3bcem8B1Vk-KcDt-oI-
bqoXkcEx2AP2iyDEfIJ7pHKRrmK8LouOwc9d0ImKhDf1hSkCqiqBKCg__&Key-Pair-
Id=APKAJGEIJM35VM43NQJA",
     "contentType": "image/jpg",
     "contentLength": 76341
   }
   "uplink_id": "1111aaaa-4222-bbbb-3333-cccc4444dddd",
    "date": "2018-10-23T00:00:00+09:00"
}
```

3.7.4. Webhookの形式(データ形式、ELTRESデバイスについて)

ELTRESデバイスからwebhookにて受け渡されるJSONの定義

ELTRESデバイスからBraveGATE CORE経由でwebhook機能(POST)にて渡されるデータはJSON形式であり、下記の仕様となります。

{ "version": <formatバージョン番号 ELTRESの場合は1>, "uplink id": "<アップリンク固有ID(UUID形式)>", "timestamp": <データがデバイスから送信された時間(UTC epoch 秒表記,UNIX TIME)>, "device": { "device id": "<センサーデータを送信したdevice id>", "group_id": "<登録デバイスが所属しているgourp id>", "battery": <バッテリ残量(%)>, "rssi": <受信強度を表す指標値(整数) RSSIが取得できない場合は null>, "sensor_id": "<センサー番号>", "sensor name": "<センサー名>", "data": {/* 各センサーにより内容が異なります。*/ "センサーを表す名前のkey": <センサー値> } } }

ELTRESの受信強度を表す指標値について

送信機が放射した電波をELTRESの受信設備で受信した際の電波強度を、0~18までの19段階で表したもの となります。数値が大きいほど電波が強かったことになります。 ただし、電波の性質上、サービスエリア 内の同一地点においても電波状態は刻々と変動する為、受信しづらくなる場合があります。

Table 1. 受信強度を表す指標値について

RSSI指標值	説明
5~18	極めて良好な電波強度
3~4	良好な電波強度で比較的安定に通信できた値
1~2	電波強度は弱いが一応の通信が可能であった値
0	非常に電波強度が弱く受信限界であった値

例) ELTRESのGSPセンサーのJSONデータ

例としてGPSセンサーのデータ形式を上げます。基本的には

- .device.sensor_id
- .device.sensor_name
- .device.data

のキーの内容が、各デバイスのセンサーの種類によって異なります。 詳しくは、各デバイスのAPIコマンド仕様書を参照下さい。

```
{
 "version": 1,
 "uplink_id": "1111aaaa-4222-bbbb-3333-cccc4444dddd",
 "timestamp": 1642646182,
 "device": {
   "device_id": "246880ffff000004",
   "battery": 50,
   "rssi": 6,
   "sensor id": "000b",
   "sensor_name": "GPS",
   "data": {
     "latitude": 33.57609558105469,
     "longitude": 130.25271606445312
   }
 }
}
```

3.7.5. アプリケーション(データ受信先のAzure Blob Storage)の設定 (POST /applications)

デバイスのセンサーデータを受け取るAzureBlobStorageのURL、コンテナー名を、POST /v1/applications にて設定します。AzureBlobStorageへのアクセスは共有アクセス署名(Shared Access Signature: 以下、SAS)を使用して行います。また、AzureBlobStorageのSAS URLは、group id毎に指定 できます。

```
$ curl -s https://api.braveridge.io/v1/applications -X POST -H X-Braveridge-API-Key:
@AzureBlobJson.json
<RESPONSE>
{
 "settings": {
  "sas_url": "https://example.com/hook/aaabbbccc",
  "container_name": "example-container",
  "prefix" : "sample/"
 },
 "updated_at": "2018-12-17T18:43:54+09:00",
 "application_type": "azure_blob_storage",
 "name": "put azure blob storage for thermal sensor",
 "created_at": "2018-12-17T18:43:54+09:00",
 }
```

 登録したアプリケーションのIDをappication_idとして返します。このIDを利用して、詳細情報の取 得、更新、削除に使用します。

ファイル内容: azure_blob_storage.json

 container_nameに指定できる文字や文字数は、Azure blob storageサービスのコンテナー名の命名規 則に準拠します。

2 prefix は、jqコマンドの構文で指定することができます。省略可。

③ group_idは、optional。指定しなければ、default group となります。

アプリケーション登録時の接続チェックについて

アプリケーションのSAS_URLを利用して、指定されたコンテナーにデータを保存可能かチェックします。 接続チェックの際には指定されたコンテナーに以下のようなファイルが作成されます。

・チェック用のファイル: braveridge_validations

This file was created by the BraveGATE API. https://api.braveridge.io/v1/applications

(C)Braveridge co., ltd. https://www.braveridge.com/

prefixの指定について

prefix を指定することにより、センサーデータのパスをカスタムすることができます。prefix にはjqコマ ンドの記述でUplink Jsonからデータを抽出し、指定することができます。

例として`prefix`の指定とセンサーのJSONデータをあげます。

・例) prefix の指定

"prefix":

"{.timestamp|strftime("%Y/%m/%d/%H/")}{.timestamp}_{.device.device_id}_{.device.sensor _id}_{.uplink_id}.json"

・例)センサーのJSONデータ

```
{
    "group": {
        "group_id": "GP0123456789abcdefghijkABCDEFGHIJK",
        "name": "Test Group"
    },
    "application": {
        "application_id": "AP0123456789abcdefghijkABCDEFGHIJK",
        "name": "Test Put Azure Blob Storage"
    },
    "router": {
        "router_id": "bb000001",
        "imsi": "440000000000001",
        "rssi": -50,
        "battery": 75,
        "fw version": "0.0.1"
    },
    "device": {
        "device_id": "2468800004000004",
        "sensor_id": "0004",
        "sensor_name": "Temperature sensor",
        "rssi": -50,
        "data": {
            "temperature": 12.34
        }
    },
    "uplink_id": "1111aaaa-4222-bbbb-3333-cccc4444dddd",
    "date": "2018-10-23T00:00:00+09:00",
    "timestamp": 1540220400
}
```

この場合のセンサーデータは 2018/10/22/15/1540220400_2468800004000004_0004_1111aaaa-4222-bbbb-3333-cccc4444ddd.json にて保存されます。

NOTE strftime()による日付フォーマットの指定は、UNIXTIMEで出力します。

3.7.6. Azure Blob Storageへ保存するデータ形式

Webhookと同様にJSON形式で保存されます。詳細はWebhookの形式(データ形式)を参照してください。

3.7.7. アプリケーションの情報一覧 (GET /applications)

現在登録されているされているアプリケーションの一覧を GET /v1/applications にて取得します。

例) アプリケーションの一覧を取得

```
$ curl -s https://api.braveridge.io/v1/applications -X GET -H X-Braveridge-API-Key:
<RESPONSE>
{
  "applications": [
     {
        "name": "webhook for thermal sensor",
        "application_type": "webhook",
        "settings": {
          "url": "https://example.com/hook/aaabbbccc"
        },
        "created at": "2018-12-17T18:43:54+09:00",
        "updated_at": "2018-12-17T18:43:54+09:00"
     }
  ],
  "total": 1,
  "pages": 1,
  "limit": 10,
  "current_page": 1,
  "next_page": null
}
```

3.7.8. アプリケーション詳細取得 (GET /applications/{application_id})

BraveGATE COREに登録されたアプリケーションの個別の情報を GET /v1/applications/{application_id} にて取得します。

例) アプリケーションの詳細情報を取得

3.7.9. アプリケーションの情報更新(PUT /applications/{application_id})

アプリケーションの名前や所属させるグループ、webhookを通知するURLを変更する場合には、 PUT /applications/{application_id} にて情報の更新を行います。

```
例) アプリケーションの詳細情報を取得
```

```
-X PUT -H X-Braveridge-API-Key: AKOAqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-
@application_put.json
<RESPONSE>
{
 "settings": {
  "url": "https://new.example.com/hook/dddeeeffff"
 },
 "updated_at": "2019-10-10T10:31:35+09:00",
 "group id": "GRec3b8f3YYYYYYYYYYYYYYYYYYYYYYYYYYYYY
 "application type": "webhook",
 "name": "new_webhook for thermal sensor",
 "created_at": "2018-12-17T18:43:54+09:00",
 }
```

BraveGATE CORE ユーザーズガイド



ファイル内容: application_put.json

① 変更したいキーを指定します。

3.7.10. アプリケーション削除(DELETE /applications/{application_id})

アプリケーションの登録をBraveGATE COREから削除します。

例) アプリケーションを削除

3.8. コマンド送信 (Downlink)

BraveGATEでは、登録したルーター・デバイスに対してWEB APIにて容易にコマンドを送信する事を実現しています。以下に代表的なコマンドの例を記載します。(個別のデバイス、ルーターのバージョンによって実装されているコマンドが異なります。詳しくは、個別のルーター、デバイスのマニュアルをご参照下さい。)

Table 2. Device

コマンド	対象	説明
SEND_DATA_AT_ONCE	Device	即時Uplink要求
SET_UPLINK_INTERVAL	Device	Uplink間隔設定
SET_UPLINK_TERM	Device	Uplink期間設定(指定期間後にアップリンク)
SET_UPLINK_THRESHO LD	Device	Uplink閾値設定
SET_SENSOR_RANGE	Device	測定値範囲設定(指定範囲にセンサー値が示した場 合にアップリンク)
SET_REGISTER	Device	デバイスのレジスタ値を設定
DOWNLOAD_DATA	Device	データダウンロード
SET_UPLINK_ENABLE	Device	Uplink有効設定
SET_SENSOR_ENABLE	Device	センサー有効設定
UPDATE_DEVICE_FIRM WARE	Device	デバイスDFU
RESTART	Device	デバイス再起動
POWER_OFF	Device	デバイスPowerOff

Table 3. Router

コマンド	対象	説明
UPDATE_ROUTER_FIR MWARE	Router	ルーターDFU
SET_FILTER_DEVICE	Router	Scan対象デバイス設定(device_id指定)
CLEAR_FILTER_DEVICE	Router	Scan対象フィルター設定クリア(device_id指定)
SCAN_DEVICE	Router	周辺デバイス検索
CHECK_ALIVE	Router	ルーター生存確認
RESTART	Router	ルーター再起動
POWER_OFF	Router	ルーターPowerOff

(詳細な動作仕様は、機器の仕様書を参照下さい。)

3.8.1. コマンド送信(POST /commands)

ルーター、デバイスにJSON形式でコマンドパラメータ送信します。パラメータを指定してルーター、デバ イスの動作を制御することが可能です。パラメーターの仕様については、各ルーター、デバイスのAPI仕 様書を参照下さい。

例) デバイスからのUplinkの閾値を設定

ファイル内容: downlink.json

```
{
    "name": "SET_UPLINK_THRESHOLD",
    "params": {
        "sensor_id": "0018", ①
        "threshold": 1000 ②
    },
    "targets": {
            "devices": [
               "24688000xxxxxxxx"
        ]
    }
}
```

① 制御するセンサーのIDを指定

2 閾値1000 になったら、データーをアップリンク

コマンド送信は、非同期処理となるため(HTTP のレスポンスは、202 Acceptedを返答し ます。)要求処理は受け入れられましたが、実際のルーター、デバイスへの処理は完了し ていません。要求は、実際に処理される時に拒否される可能性があるため、最終的に処理 NOTE されない場合もあります。

> クライアントは、レスポンスに含まれるジョブ・リソースの Location ヘッダーを調べ て、ジョブが終了した時期を確認する必要があります。コマンド結果の詳細の取得を参照

次に、コマンド結果の状態を取得する方法について記載します。

3.8.2. コマンド結果一覧の取得(GET /commands)

現在実行されているコマンドや直近に実行したコマンドの送信結果の一覧は、 GET /v1/commands で確認 します。

NOTE コマンドの結果情報は、実行から一定期間(48時間)経過したものから順にサーバーから消去されます。

例) コマンド結果一覧の取得

```
$ curl -s https://api.braveridge.io/v1/commands -X GET -H X-Braveridge-API-Key:
<RESPONSE>
{
 "commands": [
   {
     "command_id": "CDN1Hz282gPZDmwDubS9bAifWc1XUXFhXG",
"url":"https://api.braveridge.io/v1/commands/CDN1Hz282gPZDmwDubS9bAifWc1XUXFhXG",
     "name": "SET_UPLINK_THRESHOLD",
     "accepted_at": "2018-12-17T18:43:53+09:00",
     "status_counts": {
      "processed": 0,
      "rejected": 0,
      "pending": 1,
      "queued": 0
    }
   }
 ],
 "total": 1,
 "pages": 1,
 "limit": 10,
 "current_page": 1,
 "next_page": null
}
```

コマンドレスポンスの定義

サーバーから返されるコマンド結果一覧のレスポンスbodyは、JSON フォーマットであり、下記の仕様となります。

```
{
 "commands": [ /* 過去24時間に実行されたコマンド実行の実行状態の概要一覧 */
  {
    "command id": "<実行したコマンドのcommand id>",
    "url":"<コマンド詳細のURL
が記載、https://api.braveridge.io/v1/commands/{command_id}>",
    "name": "<指定したコマンド名>",
    "accepted at": "<コマンドが受付された日時>",
    "status counts": {
      "processed": <コマンドの実行が成功した個数>,
      "rejected": <エラーにてコマンドが実行できなかった個数>,
      "pending": <コマンド送信待ちの個数>,
      "queued": <ポーリングダウンリンクモードのコマンドがqueingされている個数>
    }
  }
 ],
 "total": <一覧表示するcommands がトータルで何件あるかの配列の要素数>,
 "pages": <全体のページ数>,
 "limit": <1ページに表示する最大数>,
 "current_page": <現在表示しているページ>,
 "next_page": <次のページのURL>
}
```

コマンド一覧レスポンスにて、ターゲットに指定したデバイスの現在の個々のステータス状況を status_counts にて合計して示します。複数のルーター、デバイスを指定した場合には、指定したルータ ー、デバイス数の状況をカウントします。グループで指定した場合は、グループに所属するルーター、デ バイス数の分がカウントされ表示されます。

Table 4. status_counts

Key	説明
pending	コマンドのdownlink送信待ち、もしくは送信中にて応答(Downlink Response)待ちの状態の合計数
rejected	バリデーションエラー、コマンド拒否(ルータへの送信失敗)、コマンド実行 で失敗した合計数
processed	コマンドの実行が成功した合計数(一部の命令によっては、命令の受信が成功 したの場合もあり。)
queued	ポーリングダウンリンクモードの経路にてサーバー側にてキューイングされて いる合計数

ポーリングダウンリンクモードは、一部ルーター(一体側デバイス等)にて対応しています。このモードは、デバイスが常時通信状態になく、そのままではダウンリンクの命令を受け取れないデバイスにダウンリンク命令を送信するための仕組みです。BraveGATE COREは、デバイスがどのダウンリンクモードかを判別して最大100個までの命令を溜めておきます。また最大のコマンドの滞留期間は48時間となります。デバイス側は任意のタイミイングにてBraveGATE CORE 側に自分宛の命令がないか問い合せを行ない、自分宛の命令が存在すれば溜まった命令を受け取ります。ポーリングモードの詳細な動作仕様については、各デバイスの仕様書をご参照下さい。

3.8.3. コマンド結果の詳細の取得(GET /commands/{command_id})

コマンドを送信すると、RESTの仕様に従いレスポンスヘッダー Location には実行したコマンドに割り当 てられたリソースのURIが設定されます。このリソースのURIを利用し、コマンド結果の詳細情報を取得し ます。

NOTE コマンドの結果情報は、実行から一定期間(48時間)経過したものから順にサーバーから消去されます。

コマンド結果に関しては、非同期要求-応答パターンにて実装されているため、各コマンドの結果について は、非同期ポーリングにて発行したコマンドの結果を取得する必要あります。

例) コマンドを実行した場合のレスポンスヘッダー

① location に設定されたリソースのURIを使用すると、実行したコマンド結果の詳細を取得できます。

例) locationに設定されたリソースURIを指定してコマンド結果の詳細を取得

```
$ curl -s https://api.braveridge.io/v1/commands/CDNKTOBRxtNVIs3YfLYmagKU4DkdPKWkAi -X
{
 "command_id": "CDNKTOBRxtNVIs3YfLYmagKU4DkdPKWkA",
 "accepted_at": "2019-01-30T21:02:53+09:00",
 "params": {
   "sensor id": "0060",
   "threshold": 1000
 },
 "status": "running",
 "name": "SET UPLINK THRESHOLD",
 "targets": {
   "devices": [
    {
      "device_id": "24688000xxxxxxx",
      "router_id": "1234xxxx",
      "status": "pending",
      "sent_at": "2018-12-17T18:43:54+09:00",
      "histories": [ ①
         {
             "sent_at": "2018-12-17T18:40:46+09:00",
             "reason": "device unreachable.",
             "router id": "4321xxxx",
             "status": "rejected"
         }
      ]
    }
  ]
}
}
```

 BraveGATEは、デバイスの周辺に存在するルーターを優先度に従って2台まで選択を行いリトライ処理 を行ないます。リトライ処理が行なわれたときのみ、コマンドの実行履歴を表示します。target が device で、周辺にrouterが複数置かれている場合のみ。

コマンド結果の詳細のレスポンスJSON仕様について

サーバーから返されるコマンド結果詳細のレスポンス body は JSON フォーマットであり、下記の仕様となります。

```
{
   "command_id": "<実行したコマンドのcommand id>",
   "accepted_at": "<受付された日時>",
   "parmas": {
      <commands 送信時に指定したパラメーター>
   },
   "status": "cheking|accepted|running|completed",
   "name": "<指定したコマンド名>"
   "target": { // ターゲットにて指定した、devices,routers,groupsが配列にて表記
.statusがcheking時にはNULL
      "devices": [
         {
            "device_id": "<指定したdevice_id>",
            "router_id": "<コマンドを中継したrouter id>",
            "status": "<pending|processed|reject|queued>",
            "reason": "<rejectにて実行が失敗した場合に失敗理由を記載>",
            "sent_at": "<コマンドがルーターに送信された日時(ISO 8601)>" //
BraveGATE側でバリデーションなどのエラーが発生した場合は、表示されません
            "histories": [ // リトライが発生した場合のみ表示
            {
                "sent at": "<コマンドがルーターに送信された日時(ISO 8601)>",
                "reason": "<rejectにて実行が失敗した場合に失敗理由を記載>",
                "router_id": "<コマンドを中継したrouter_id>",
                "status": "rejected"
            }
      ],
      "routers": [
         {
            "router id": "<指定したrouter id>",
            "status": "pending|processed|reject|queued",
            "reason": "<rejectにて実行が失敗した場合に失敗理由を記載>",
            "sent at": "<コマンドがルーターに送信された日時(ISO 8601)>"
         }
      1
  }
}
```

詳細は、http://developer.braveridge.io/docs/api/v1/#/Command/getCommand を参照ください。

コマンド実行結果状況 status key について

コマンド詳細のレスポンスにて表示される、status key にて現在のコマンドの実行状況を表示します。コマンドの実行結果を最終的に取得する場合には、該当コマンドの status が completed になるまでポーリングして確認する必要があります。

Table 5. status

ステータス	説明
checking	コマンド内容確認中(バリデーション等)

ステータス	説明
accepted	バリデーションチェックを行なった後、コマンドを受け付けた。
running	コマンド処理待ち、処理中。ルーターへのdownlink送信を待機中、もしく は、応答(Downlink Response)待ちの状態
completed	コマンドの処理にて、成功、失敗となんらかの結果が判明し、処理を完了。

コマンド target status一覧

コマンド詳細のレスポンスにて表示される、個々のtarget
target.<devices|routers|groups>[配列].status
にて現在のコマンドの実行状況を表示します。

NOTE 最終的な実行は、コマンドの実行が成功の場合は、processed となり、失敗の場合は rejected となります

Table 6. status

ステータス	説明
pending	コマンドのdownlink送信待ち、もしくは送信中にて応答(Downlink Response)待ちの状態
rejected	バリデーションエラー、コマンド拒否(ルータへの送信失敗)、コマンド実行 失敗
processed	コマンドの実行が成功し処理済みとなっている。(一部の命令によっては、命 令の受信が成功した状況を表す場合があります。各デバイス、ルーターのAPI 仕様書等を確認下さい。)
queued	ポーリングダウンリンクモードの経路に対してルーター側が問い合せに来るま で、BraveGATE側にてキューイングされている状態

3.8.4. コマンド実行時のエラーケースについて

BraveGATEにてダウンリンクコマンドを実行する際に、エラーが起こる箇所について図で示します。



(1) BraveGATE COREがコマンドを受付、ルーター・デバイスの命令 (バイナリー命令)に変換しルーター へ送信する前に起こるエラー

(2) BraveGATE COREからルーターへ命令を閉域網を通して送信する時点で起こるエラー

(3) 受信したコマンドをデバイスに中継する時点で起こるエラー

各エラーケースについて、reasenの一覧

コマンドが失敗する(rejected のsutatusになる)場合に、 reasen keyに表記されるエラーについて、エラ ーケース事に一覧を示します。

Table 7. (1) BraveGATE COREがコマンドを受付、ルーター・デバイスの命令 (バイナリー命令)に変換しルーターへ送信する前に起こるエラー

type	reasen	エラー内容詳細
group	group_id is invalid.	target にgroupを指定した場合に、指定した gorup id が不 正な値。文字長が34文字ではない。group idの先頭文字 がGRから開始されていない。
group	group not found.	存在しないgroup idを指定した。アカウントに紐付けられ たgroup idではない。
router	router_id is invalid.	router id が不正。文字長が8文字ではない。
router	router not found.	ルーターが登録されていない。もしくはrouter idに保有し ていない誤ったidを指定した。
router	command not supported.	ルーターで実行できないコマンドを指定した。sensor id で 指定したセンサーデバイスが定義されていない。
device	device_id is invalid.	device idが不正。device id の文字長が16文字ではない。

type	reasen	エラー内容詳細
device	router not found.	デバイスにダウンリンクを送信できるルーターの情報 がBraveGATE上に存在しない。また、送信できるルーター の情報が存在したとしても、該当するルーターが存在しな い。誤ったrouter id を指定した。 デバイスがルーター対 して一度も接続していない。
device	sensor not supported.	sensor id が指定された場合に、指定されたセンサーデバイ スがBraveGATE CORE上で定義されてない。
device	interval must be 10~65535.	SET_UPLINK_INTERVALを指定したときに、paramで指定 した値が10から65535の範囲にない。
device	term must be 10~65535.	SET_UPLINK_TERMを指定したときに、paramで指定した 値が10から65535の範囲にない。
device	reg_len must be 0~1500.	REG_SETTINGを指定したときに、reg_len の値の長さが0 以上-1500以下ではない
device	reg_bin must be 0~1500.	REG_SETTINGを指定したときに、reg_bin の長さが0 かNULL。
device	[sensor name] must be [数 値]~[数値]	SET_UPLINK_THRESHOLDを指定したときに、センサーで サポートされている範囲外の値が指定された。

Table 8. (2) BraveGATE COREからルーターへ命令を閉域網を通して送信する時点で起こるエラー

type	reasen	エラー内容詳細
router	router is busy.	ルーター(半二重通信の機器)がアップリンク中で、ダウン リンクコマンドwを受け付けない。
router	router timeout.	接続を確立した後、通信の途中でタイムアウト(ルータ側も しくはデバイス側から応答がない。BraveGATE CORE 側 は、通信タイムアウト90sec)
router	router unreachable.	LTE網経由でのIP通信が行えない。ルーターへの通信が到 達しない。
router	router refused.	LTE網経由でのIP通信は行えるが、ルーターと接続が確立 できない。

Table 9. (3) 受信したコマンドをデバイスに中継する時点で起こるエラー

type	reasen	エラー内容詳細
device	sensor not supported.	エンドデバイスにサポートされていないsensor idを指定し て命令を実行。エンドデバイスとBraveGATE CORE側の定 義の不一致。(デバイス開発中にて、未実装段階でのエラー 等)
device	command not supported.	エンドデバイスにて未サポートのコマンドが実行された。 エンドデバイスとBraveGATE CORE側の定義の不一致。(デ バイス開発中にて、未実装段階でのエラー等)

Braveridge

type	reasen	エラー内容詳細
device	value is invalid.	エンドデバイスにて設定範囲外のパラメーターを指定し た。エンドデバイスとBraveGATE CORE側の定義の不一 致。(デバイス開発中にて、未実装段階でのエラー等)
router	device unreachable	ルーターからエンドデバイスに接続しようとしたが接続で きなかった。Bluetooth にて、周辺デバイスの探索を行っ たが発見できなかった。等 *ルーターからエラーを返答
device	device timeout.	エンドデバイスとの接続を確立したが、データを送信中に タイムアウトが発生。*ルーターからエラーを返答
device	unknown error.(X)	不明なエラー *ルーターからエラーを返答

3.9. モバイル通信量の取得 (BraveSIM)

現在使用しているSIMの通信量を GET /v1/sims/{imsi} にて取得します。 SIMに割り当てられてい るIMSI(加入者識別番号)を、パラメータとして指定し、 該当するSIMの通信量を取得します。過去6ヶ月 分の通信量を取得する事ができます。

NOTE 12:00 (JST) までに、前日(JST)のパケット通信料が集計されます。

3.9.1. IMSIの確認方法

機器が使用しているSIMのIMSIの確認は、ルーター詳細取得(GET /routers/{router_id}) にて確認する事ができます。 詳細は、ルーター詳細取得 を参照下さい。

例) ルーター詳細取得の内容

① imsi キーの、15桁の数字を確認します。

3.9.2. モバイル通信量の取得 (GET /sims)

クエリパラメーターを指定しない場合は、当月の1日から前日までの日毎の通信量を表示します。

例) 2020年6月10日13:00 にSIMの通信量を取得した場合

```
$ curl -s https://api.braveridge.io/v1/sims/440012345678901 -X GET -H X-Braveridge-
<RESPONSE>
{
   "data_usages": [
      {
         "date": "20200601",
         "uplink_byte": 1094,
         "downlink_byte": 584
      },
      {
         "date": "20200602",
         "uplink_byte": 462141,
         "downlink_byte": 315264
      },
      ~省略~
      {
         "date": "20200608",
         "uplink_byte": 224550,
         "downlink_byte": 153640
      },
      {
         "date": "20200609",
         "uplink_byte": 1391827,
         "downlink_byte": 955614
      }
   ],
   "total_downlink_byte": 6322351,
   "total_uplink_byte": 9155993
}
```

start、 end の期間を指定するクエリパラーメターを指定した場合には、指定した期間の通信量を取得し ます。(start,end の両方の日付を指定する必要があります。) 例) 期間を指定して(2020/1/1~2020/1/31)SIMの通信量を取得した場合

```
$ curl -s
https://api.braveridge.io/v1/sims/440012345678901?start=20200101&end=20200131 -X GET
<RESPONSE>
{
   "data_usages": [
      {
         "date": "20200101",
         "uplink_byte": 1094,
         "downlink_byte": 584
      },
      {
         "date": "20200102",
         "uplink_byte": 462141,
         "downlink_byte": 315264
      },
      ~省略~
      {
         "date": "20200130",
         "uplink_byte": 224550,
         "downlink_byte": 153640
      },
      {
         "date": "20200131",
         "uplink_byte": 1391827,
         "downlink_byte": 955614
      }
   ],
   "total_downlink_byte": 6322351,
   "total_uplink_byte": 9155993
}
```

NOTE パケット消費量の料金計算は、上りと下りを合算して行われます。

3.9.3. router id を利用したモバイル通信量の取得 (GET /routers/{router_id}/data_usage)

router idを利用してルーターに搭載されているSIMの通信量を取得する事ができます。 クエリパラメータ ーを指定しない場合は、当月の1日から前日までの日毎の通信量を表示します。 例) 2020年6月10日13:00 にSIMの通信量を取得した場合

```
$ curl -s https://api.braveridge.io/v1/routers/bb000001/data_usage -X GET -H X-
<RESPONSE>
{
   "data_usages": [
      {
         "date": "20200601",
         "uplink_byte": 1094,
         "downlink_byte": 584
      },
      {
         "date": "20200602",
         "uplink_byte": 462141,
         "downlink_byte": 315264
      },
      ~省略~
      {
         "date": "20200608",
         "uplink_byte": 224550,
         "downlink_byte": 153640
      },
      {
         "date": "20200609",
         "uplink_byte": 1391827,
         "downlink_byte": 955614
      }
   ],
   "total_downlink_byte": 6322351,
   "total_uplink_byte": 9155993
}
```

start、 end の期間を指定するクエリパラーメターを指定した場合には、指定した期間の通信量を取得し ます。(start,end の両方の日付を指定する必要があります。) 例) 期間を指定して(2020/1/1~2020/1/31)SIMの通信量を取得した場合

```
$ curl -s
https://api.braveridge.io/v1/routers/bb000001/data_usage?start=20200101&end=20200131
<RESPONSE>
{
   "data_usages": [
      {
         "date": "20200101",
         "uplink_byte": 1094,
         "downlink_byte": 584
      },
      {
         "date": "20200102",
         "uplink_byte": 462141,
         "downlink_byte": 315264
      },
      ~省略~
      {
         "date": "20200130",
         "uplink_byte": 224550,
         "downlink_byte": 153640
      },
      {
         "date": "20200131",
         "uplink_byte": 1391827,
         "downlink_byte": 955614
      }
   ],
   "total_downlink_byte": 6322351,
   "total_uplink_byte": 9155993
}
```

NOTE パケット消費量の料金計算は、上りと下りを合算して行われます。

4. ダウンリンクコマンド凡例

BraveGATE COREは、ルーター、デバイスに対してダウンリンクにて命令を送信する機能を提供します。 この章では、APIに送信するJSON形式のコマンドパラメータについてコマンドの凡例を説明します。デバ イスによって実装しているコマンドが異なっている為、詳細は各デバイスの仕様をご確認下さい。

各IDのサイズについては以下と定義されています。

値名	サイズ	説明
device id	英数字16桁	deviceに割当たっている固有のID
router id	英数字8桁	routerに割当たっている固有のID
sensor id	英数字4桁	sensorの種別に割当たっている固 有のID
group id	GRを接頭語とする英数字32桁	group に割当たる固有のID

大文字小文字は、区別されます。

4.1. センサデータの即時取得

指定したデバイスのセンサー情報を即時アップリンク命令により取得します。コマンドを受け取ったデバ イスは、センサーデータを取得を開始し取得後にセンサーデータをアップリンクします。

Table 10. リクエストパラメーター

JSON Key	值	必須	説明
name	String	\odot	SEND_DATA_AT_ONCE
target	Object	\odot	コマンドの送り先
devices	Array(String)	∘(devices もしくは groups)	送り先のdevice id 複数 指定可
groups	Array(String)	∘(devices もしくは gourps)	送り先のgroup id 複数指 定可
params	Object	0	
sensor_id	String	\odot	値を取得したいsecsor id

例) バッテリー情報取得

```
{
    "name": "SEND_DATA_AT_ONCE",
    "targets": {
        "devices": [
            "2468800060010020"
      ]
    },
    "params": {
            "sensor_id": "0032"
    }
}
```

4.2. センサデータの**Uplink**間隔設定

指定したデバイスのセンサー情報を定期的にアップリンクする時間間隔を指定します。

JSON Key	値	必須	説明
name	String	\odot	SET_UPLINK_INTERVAL
target	Object	\odot	コマンドの送り先
devices	Array(String)	∘(devices もしくは groups)	送り先のdevice id 複数 指定可
groups	Array(String)	∘(devices もしくは gourps)	送り先のgroup id 複数指 定可
params	Object	\odot	
sensor_id	String	\odot	値を取得するsensor id
interval	Integer	\bigcirc	uplink間隔を秒数で指 定。間隔の値はデバイス の仕様による。

Table 11. リクエストパラメーター

```
{
   "name": "SET_UPLINK_INTERVAL",
   "targets": {
        "devices": [
            "2468800001000001"
        ],
        "groups": [
            "GR0123456789abcdefghijkABCDEFGHIJK",
            ]
   },
   "params": {
        "sensor_id": "000b",
        "interval": 3600
   }
}
```

4.3. データダウンロード(デバイスのみ)

データダウンロードに対応したデバイスに対してデータをダウンリンクを行います。

JSON Key	值	必須	説明
name	String	\odot	DOWNLOAD_DATA
target	Object	\odot	コマンドの送り先指定
devices	Array(String)	∘(routers もしくは groups)	送り先のdevice id 複数 指定可
groups	Array(String)	∘(routers もしくは gourps)	送り先のgroup id 複数指 定可
params	Object	\odot	
sensor_id	String	0	データを渡したいsensor idを指定
data	String	0	BASE64エンコードした データ。サイズ は、JSONパラメーター 全体で10M byte以下

```
{
    "name": "DOWNLOAD_DATA",
    "targets": {
        "devices": [
            "246880006000001"
      ]
    },
    "params": {
        "sensor_id": "0060",
        "data": "[BASE64エンコードしたデータ]"
    }
}
```

□デバイス毎にダウンロードするデータが違う場合は1デバイス毎にPOSTする

4.4. デバイスフィルター

4.4.1. Scan対象フィルタ上書き設定(DEVICE ID)(ルーターのみ)

ルーターに対して、BLEにてSCANするデバイスを指定するフィルタを設定します。設定できるdevice_id は100個までとなります。 注) ルーターにて設定できるdeviceも100個までとなります。

Table 12. リクエストパラメーター

JSON Key	値	必須	説明
name	String	\odot	SET_FILTER_DEVICE
target	Object	\odot	コマンドの送り先指定
routers	Array(String)	∘(routers もしくは groups)	送り先のrouter id 複数 指定可(最大100)
groups	Array(String)	∘(routers もしくは gourps)	送り先のgroup id 複数指 定可
params	Object	0	
device_id	String	0	通信許可するdevice idを 指定

```
{
 "name": "SET_FILTER_DEVICE",
 "targets": {
   "routers": [
          "10000001",
          "1000002",
          "10000003"
   ],
 "groups": [
          "GR0123456789abcdefghijkABCDEFGHIJK",
          ]
 },
 "params": {
   "device_id": [
     "2468800001000001",
     "2468800001000002",
     "2468800001000003",
     "2468800001000004",
     "2468800001000005",
     ~省略~
     "2468800001000100",
   ]
 }
}
```

4.4.2. Scan対象フィルタ設定追加(DEVICE ID) (ルーターのみ)

設定されたscan対称フィルタについてフィルタを追加します。

必ずルーターにフィルターを設定した状態で実行してください。フィルターを設定していNOTE ない状態で追加の実行を行うと、ルーターとデバイスの紐付けに不整合が生じる場合があります。

Table 13. リクエストパラメーター

JSON Key	値	必須	説明
name	String	\odot	ADD_FILTER_DEVICE
target	Object	\odot	コマンドの送り先指定
routers	Array(String)	∘(routers もしくは groups)	送り先のrouter id 複数 指定可
groups	Array(String)	∘(routers もしくは gourps)	送り先のgroup id 複数指 定可
params	Object	\odot	

Braveridge

device_id	String	\bigcirc	通信許可するdevice idを
			指定

例)

```
{
 "name": "ADD_FILTER_DEVICE",
 "targets": {
   "routers": [
         "10000001",
         "1000002",
         "10000003"
   ],
   "groups": [
         "GR0123456789abcdefghijkABCDEFGHIJK",
         ]
 },
 "params": {
   "device_id": "2468800001000001"
 }
}
```

4.4.3. Scan対象フィルタ設定クリア(DEVICE ID) (ルーターのみ)

ルーターに設定されたscan対称のフィルタをクリアします。Groupを指定することにより、

JSON Key	値	必須	説明
name	String	\odot	CLEAR_FILTER_DEVICE
target	Object	\odot	コマンドの送り先
routers	Array(String)	∘(router もしくは group)	送り先のrouter id 複数 指定可
routers	Array(String)	∘(router もしくは group)	送り先の

Table 14. リクエストパラメーター

{
"name": "CLEAR_FILTER_DEVICE",
"targets": {
"routers": [
"10000001",
"1000002",
"10000003"
],
"groups": [
"GR0123456789abcdefghijkABCDEFGHIJK",
"GR0000000000xxxxxxxxXXXXXXXXXXXXXXXXXXXX
]
}
}

4.5. 電源オペレーション

4.5.1. 電源オフ

指定したルーター・デバイスに電源オフを指示します。 (デバイスは、再起動の機能の有無による)

Table 15. リクエストパラメーター

JSON Key	値	必須	説明
name	String	\odot	POWER_OFF
target	Object	\odot	コマンドの送り先
routers	Array(String)	∘(router もしくは device)	送り先のrouter id 複数 指定可
devices	Array(String)	○(router もしくは device)	送り先のdevice id 複数 指定可

```
{
   "name": "POWER_OFF",
   "params": {
   },
   "targets": {
       "devices": [
           "2468800010000001",
           "246880001000002",
           "2468800010000003"
      ],
       "routers": [
           "10000001",
           "1000002",
           "10000003"
       ]
  }
}
```

4.5.2. ルーター再起動(ルーター)

指定したルーターを再起動します。

Table 16. リクエストパラメーター

JSON Key	值	必須	説明
name	String	\odot	RESTART
target	Object	\odot	コマンドの送り先
routers	Array(String)	0	送り先のrouter id 複数 指定可
params	Object	\odot	
sensor_id	String	\odot	sensor id 0000を指定

```
例) ルーター再起動
```

```
{
    "name": "RESTART",
    "targets": {
        "routers": [
            "bb000001"
        ]
    },
    "params": {
            "sensor_id": "0000"
    }
}
```