

BraveGATE CORE ユーザーズガイド

株式会社Braveridge

バージョン2020/10/01

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
3.3. Auth API、APIKey APIについて	5
3.4. グループ機能	10
3.5. ルーター管理	15
3.6. デバイス管理	19
3.7. センサーデータの取得用アプリケーションの登録 (Uplink)	23
3.8. コマンド送信 (Downlink)	29
3.9. モバイル通信量の取得 (BraveSIM)	37
4. ダウンリンクコマンド凡例	40
4.1. センサデータの即時取得	40
4.2. センサデータのUplink間隔設定	41
4.3. データダウンロード（デバイスのみ）	42
4.4. デバイスフィルター	43
4.5. 電源オペレーション	46

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 **Exmple 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ので、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

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"を指定して下さい。

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) ページネーションの表示

```
...
  "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、APIKey APIについて

3.3.1. 初期APIKeyと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 することにより初期操作に必要な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

```
$ curl -s https://api.braveridge.io/v1/auth -X POST -d @auth_first.json

<RESPONSE>
{
  "apikey": "AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "token": "TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "expire_at": "2019-01-10T12:01:23+09:00"
}
```

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、初期`apikey`と`token`を使って `authkey_id` と `authkey_secret` の発行し、以後は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、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. APIKeyの作成(POST /apikeys)

発行された `apikey` と `token` を使って `POST /apikeys` を実行し、実際にアプリケーションで使用する `authkey_id` と `authkey_secret` を作成します。

```
$ curl -s https://api.braveridge.io/v1/apikeys -X POST -d @apikey.json -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "apikey": "AKJebp9MdkLoRoIZheWLXKXEFofijrpdGG",
  "authkey_id": "AIY6qaaio8*****",
  "authkey_secret": "ASdxogbySX*****",
  "name": "foo",
  "created_at": "2019-04-02T15:25:47+09:00",
  "updated_at": "2019-04-02T15:25:47+09:00"
}
```

ファイル内容: `apikey.json`

```
{
  "name": "foo"
}
```

3.3.3. APIKey 一覧(GET /apikeys)

発行したAPIKey 一覧を取得します。なをレスポンスの一覧表示では、`authkey_id` と `authkey_secret` については一部「*」にてマスク処理表記されます。


```
$ curl -s https://api.braveridge.io/v1/apikeys -X GET -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "apikeys": [
    {
      "apikey": "AKJebp9MdkLoRoIZheWLXKXEFofijrpdGG",
      "authkey_id": "AIY6qaaio8*****",
      "authkey_secret":
"ASdxogbySX*****",
      "name": "foo",
      "created_at": "2019-04-02T15:25:47+09:00",
      "updated_at": "2019-04-02T15:25:47+09:00"
    },
    {
      "apikey": "AKUPAYcTSdUook006DdYCsTDGZfcKnfTq2",
      "authkey_id": "AIyX3iYnDe*****",
      "authkey_secret":
"ASBwoyb5V6*****",
      "name": "",
      "created_at": "2019-03-22T20:01:50+09:00",
      "updated_at": "2019-03-22T20:01:50+09:00"
    },
    {
      "apikey": "AK0u5tJhnc2IYL4Rx40VpEogG6wqHf1Jv7",
      "authkey_id": "AIkWIwiDGN*****",
      "authkey_secret":
"ASGjD9S0Sv*****",
      "name": "",
      "created_at": "2019-03-22T20:01:32+09:00",
      "updated_at": "2019-03-22T20:01:32+09:00"
    },
    {
      "apikey": "AKq069GCfpFPxm0ysCBwT1JDnyXFs6K9wx",
      "authkey_id": "AIAS9cJLuY*****",
      "authkey_secret":
"ASHSxqBzzN*****",
      "name": "",
      "created_at": "2019-03-07T21:17:16+09:00",
      "updated_at": "2019-03-07T21:17:16+09:00"
    },
    {
      "apikey": "AK0AqX9S3Fgw2GcKCH1MyfXezSyjNKOPHn",
      "authkey_id": "AIGPCYmgds*****",
      "authkey_secret":
"ASRoB5g7bu*****",
      "name": "BB API",
      "created_at": "2018-11-22T12:29:17+09:00",
```

```

        "updated_at": "2018-11-22T12:29:17+09:00"
      }
    ],
    "total": 5,
    "pages": 1,
    "limit": 10,
    "current_page": 1,
    "next_page": null
  }

```

3.3.4. APIKey削除(DELETE /apikeys/{apikey})

`apikey` を指定し、`DELETE /apikeys/{apikey}` にて APIKeyを削除し無効にします。

例) APIKeyを削除

```

$ curl -s https://api.braveridge.io/v1/apikeys/AKJebp9MXXXXXXXXXXXXXXXXXXXXXXXXX -X
DELETE -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-
Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

3.3.5. APIアクセス用Tokenの発行(POST /auth)

作成された `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: ASCdE0f1ghi2j3KLMNoPqRStuV45wxYZ6aBcDEfGHIJk8lmN0pqr9S0Tuv1wXyzabC

```

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。

実行コマンド

```

$ curl -s https://api.braveridge.io/v1/auth -X POST -d @auth.json

<RESPONSE>
{
  "apikey": "AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX",
  "token": "TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "expire_at": "2019-01-10T12:01:23+09:00"
}

```

ファイル内容: *auth.json*

```
{
  "authkey_id" : "ABCDEFGHijklmnopqrstuvwxyzabcdefgh",
  "authkey_secret" :
  "ABCdE0f1ghi2j3KLmNoPqRStuV45wxYZ6aBcDEfGHIJk8lmN0pqr9S0Tuv1wXyzabC",
  "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。

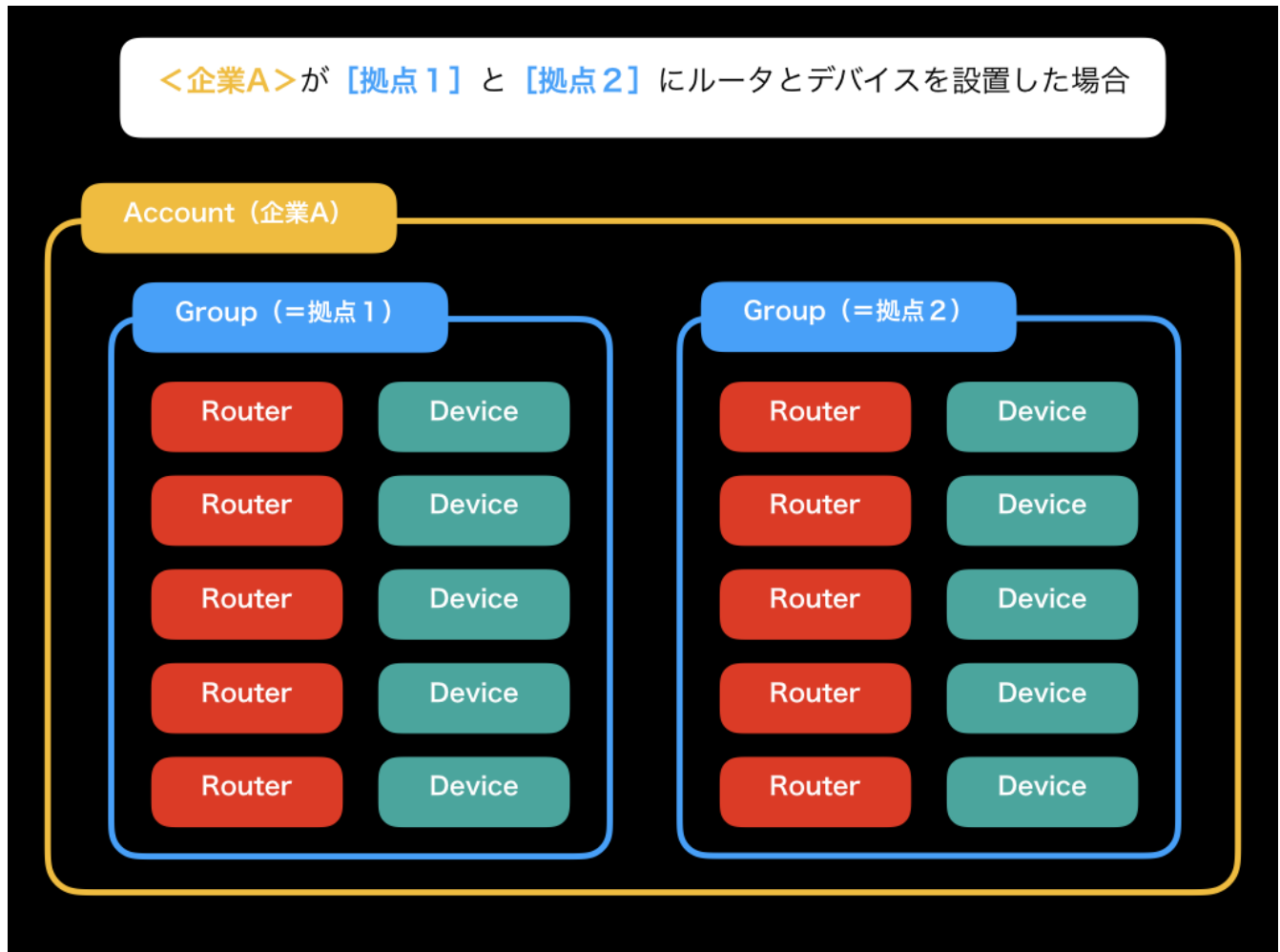
例) *key,token*を設定して実行コマンド

```
$ curl -s https://api.braveridge.io/v1/command -X GET -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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。



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

```
$ curl -s https://api.braveridge.io/v1/groups -X POST -H X-Braveridge-API-Key:
AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d
@create_group.json
```

<RESPONSE>

```
{
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "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"
}
```

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) グループ一覧を取得

```
$ curl -s https://api.braveridge.io/v1/groups -X GET -H X-Braveridge-API-Key:
AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "groups": [
    {
      "group_id": "GRhaqCpyXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "name": "",
      "is_default": false,
      "created_at": "2019-10-02T14:14:16+09:00",
      "updated_at": "2019-10-02T14:14:16+09:00"
    },
    {
      "group_id": "GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "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"
    },
    {
      "group_id": "GRDfv0LuXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "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
}
```

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

```
$ curl -s https://api.braveridge.io/v1/groups/GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXX -X
GET -H X-Braveridge-API-Key: AKOAqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
<RESPONSE>
{
  "group_id": "GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXX",
  "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"
}
```

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

```
$ curl -s https://api.braveridge.io/v1/groups/GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXX -X
PUT -H X-Braveridge-API-Key: AKOAqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d
@change_group.json
```

```
<RESPONSE>
{
  "group_id": "GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXX",
  "name": "floor-2 group",
  "is_default": true,
  "created_at": "2019-10-02T14:09:57+09:00",
  "updated_at": "2019-10-03T16:11:56+09:00"
}
```

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

```
$ curl -s https://api.braveridge.io/v1/groups/GRA8NmydXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -X  
DELETE -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-  
Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```


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

```
$ curl -s https://api.braveridge.io/v1/routers -X POST -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d @router.json
```

ファイル内容: `router.json`

```
{
  "router_id": "bb054321",
  "registration_code": "abcd1234",
  "group_id": "",
  "name": "foo bar"
}
```

`group_id`を指定しなかった場合、default group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:
AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "routers": [
    {
      "router_id": "bb054321",
      "name": "foo",
      "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "imsi": "440012345678901",
      "rssi": 0,
      "battery": -1,
      "sim_status": "ready",
      "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": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "imsi": "440012345678902",
      "rssi": -69,
      "battery": 48,
      "sim_status": "ready",
      "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
}
```

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-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
<RESPONSE>
{
  "router_id": "bb054321",
  "name": "",
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXX",
  "imsi": "440012345678901",
  "rssi": -68,
  "battery": 30,
  "sim_status": "ready",
  "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

```
$ curl -s https://api.braveridge.io/v1/routers/bb054321 -X PUT -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d @router_put.json
```

ファイル内容: `router_put.json`

```
{
  "group_id": "GRec3b8f3YYYYYYYYYYYYYYYYYYYYYYYY",
  "name": "foo bar baz"
}
```

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。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

```
$ curl -s https://api.braveridge.io/v1/routers/bb054321 -X DELETE -H X-Braveridge-API-  
Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:  
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

```
$ curl -s https://api.braveridge.io/v1/devices -X POST -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXX -d @device.json
```

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、default group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

```
$ curl -s https://api.braveridge.io/v1/devices -X GET -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "devices": [
    {
      "device_id": "0505050505050505",
      "name": "",
      "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXX",
      "sensors": [
```

```

    {
      "sensor_id": "0004",
      "name": "Temperature sensor",
      "executable_commands": [ ①
        "SEND_DATA_AT_ONCE",
        "SET_UPLINK_INTERVAL",
        "SET_UPLINK_TERM",
        "SET_UPLINK_THRESHOLD"
      ]
    },
    {
      "device_id": "0606060606060606",
      "name": "",
      "group_id": "GRec3b8f31f2caef3a749353685afa7555",
      "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

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-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "device_id": "2468801234012345",
  "name": "Tracker09",
  "group_id": "GRec3b8f31f2caef3a749353685afa7555",
  "sensors": [
    {
      "sensor_id": "000b",
      "name": "GPS",
      "executable_commands": [
        "SEND_DATA_AT_ONCE",
        "SET_UPLINK_INTERVAL"
      ]
    },
    {
      "sensor_id": "001e",
      "name": "Full Color LED",
      "executable_commands": [
        "SET_SENSOR_RANGE"
      ]
    }
  ],
  "created_at": "2019-02-21T14:16:31+09:00",
  "updated_at": "2019-02-21T14:16:31+09:00"
}
```

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

デバイスのnameや所属させるgroup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

```
$ curl -s https://api.braveridge.io/v1/devices/2468801234012345 -X GET -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d @device_put.json
```

ファイル内容: `device_put.json`

```
{
  "group_id": "GRec3b8f3YYYYYYYYYYYYYYYYYYYYYYYYYYYY",
  "name": "foo bar baz"
}
```

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。

```
$ curl -s https://api.braveridge.io/v1/devices/2468801234012345 -X DELETE -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```


3.7. センサーデータの取得用アプリケーションの登録 (Uplink)

デバイスのセンサーデータは、あらかじめ設定したwebhookを受け取るアプリケーションのURLにBraveGATE COREからJSONデータがPOSTされることにより取得できます。デバイスによりデフォルトの固定されたuplink間隔による定期的な取得や、uplinkの間隔を指定した取得、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。

```
$ curl -s https://api.braveridge.io/v1/applications -X POST -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d @webhook.json
```

```
<RESPONSE>
{
  "settings": {
    "url": "https://example.com/hook/aaabbbccc"
  },
  "updated_at": "2018-12-17T18:43:54+09:00",
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXX",
  "application_type": "webhook",
  "name": "webhook for thermal sensor",
  "created_at": "2018-12-17T18:43:54+09:00",
  "application_id": "APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXX" ①
}
```

- ① 登録したアプリケーションのIDをapplication_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*

```
{
  "settings": {
    "url": "https://example.com/hook/aaabbbccc" ①
    "token": "abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789-._~/=" ②
  },
  "application_type": "webhook",
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXX", ③
  "name": "webhook for thermal sensor"
}
```

- 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 token に指定できる文字は、token68に準拠します。=については、最後にしか指定できません。省略

可。

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

NOTE

urlには、geury 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>
tokenを利用できない場合に、なんらかの認証を付加したい場合や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。

NOTE

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** パラメータを指定した場合は、このヘッダーが付与されます。

デバイスからBraveGATE CORE経由でwebhook(POST)にて渡されるデータはJSON形式です。BraveGATE COREとwebhook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)

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。

例) 温度センサーのJSONデータ

```
{
  "application": {
    "application_id": "APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
    "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"
}
```

3.7.4. 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:
AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "applications": [
    {
      "application_id": "APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "name": "webhook for thermal sensor",
      "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
      "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
}
```

3.7.5. 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
/v1/applications/{application_id}にて取得します。

GET

例) アプリケーションの詳細情報を取得

```
$ curl -s https://api.braveridge.io/v1/applications/APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-X GET -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-
Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<RESPONSE>

```
{
  "application_id": "APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "name": "webhook for thermal sensor",
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "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"
}
```

3.7.6. 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name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

```
$ curl -s https://api.braveridge.io/v1/applications/APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-X PUT -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-
Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d
@application_put.json
```

<RESPONSE>

```
{
  "settings": {
    "url": "https://new.example.com/hook/dddeefffff"
  },
  "updated_at": "2019-10-10T10:31:35+09:00",
  "group_id": "GRec3b8f3YYYYYYYYYYYYYYYYYYYYYYYYYYYY",
  "application_type": "webhook",
  "name": "new_webhook for thermal sensor",
  "created_at": "2018-12-17T18:43:54+09:00",
  "application_id": "APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
}
```

ファイル内容: `application_put.json`

```
{
  "settings": { ①
    "url": "https://new.example.com/hook/dddeeeffff",
    "token": "changetokenstring"
  },
  "application_type": "webhook",
  "group_id": "GRec3b8f3YYYYYYYYYYYYYYYYYYYYYYYYYYYY",
  "name": "new_webhook for thermal sensor"
}
```

① 変更したいキーを指定します。

3.7.7. 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

```
$ curl -s https://api.braveridge.io/v1/applications/APVrieKzXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
-X DELETE -H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-
Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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 1. 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_THRESHOLD	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_FIRMWARE	Device	デバイスDFUモード
RESTART	Device	デバイス再起動
POWER_OFF	Device	デバイスPowerOff

Table 2. Router

コマンド	対象	説明
UPDATE_ROUTER_FIRMWARE	Router	ルーターDFUモード
SEND_ROUTER_INFO	Router	RouterInfo要求
SET_FILTER_DEVICE	Router	Scan対象デバイス設定 (DEVICE ID)
CLEAR_FILTER_DEVICE	Router	Scan対象フィルター設定クリア(DEVICE ID)
SCAN_DEVICE	Router	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

例) デバイスからのUplinkの閾値を設定

```
$ curl -s https://api.braveridge.io/v1/commands -X POST -H X-Braveridge-API-Key:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d @downlink.json
```

ファイル内容: *downlink.json*

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

NOTE

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。ジョブの現在の状態を表すエンティティも返されます。

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(24時間)経過したのから順にサーバーから消去されます。

例) コマンドを実行した場合のレスポンスヘッダー

```
$ curl -s https://api.braveridge.io/v1/commands -X POST -H X-Braveridge-API-Key:
AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX -d "{ \"name\":
\"SEND_DATA_AT_ONCE\", \"params\": {\"sensor_id\": \"0051\" }, \"targets\": {
\"devices\": [ \"2468800051001234\" ] } }"

<RESPONSE HEADER>
HTTP/2 202
content-type: application/json
content-length: 0
location: https://api.braveridge.io/v1/commands/ CDNKTOBRxtNVI3YfLYmagKU4DkdPKWkAi ①
date: Thu, 31 Jan 2019 12:01:49 GMT
.....
```

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/CDNKTOBRxtNVI3YfLYmagKU4DkdPKWkAi -X
GET -H X-Braveridge-API-Key: AKOaqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

{
  "command_id": "CDNKTOBRxtNVI3YfLYmagKU4DkdPKWkA",
  "accepted_at": "2019-01-30T21:02:53+09:00",
  "params": {
    "sensor_id": "0060",
    "threshold": 1000
  },
  "command_seq": 0,
  "name": "SET_UPLINK_THRESHOLD",
  "targets": {
    "devices": [
      {
        "device_id": "24688000xxxxxxx",
        "status": "pending",
        "sent_at": "2018-12-17T18:43:54+09:00"
      }
    ]
  }
}
```

コマンド結果の詳細のレスポンスJSON仕様について

```
<CommandDetailResponse> {
  name      string // SEND_DATA_AT_ONCE
  params    /* CommandDetailResponse_params */ { // params
を指定していた場合(optional)
    sensor_id  string // paramsで指定したセンサーID (0パディング、数字4桁)
    interval   integer // paramsで指定した定期アップリンク等の間隔の時間(sec)
  }
  accepted_at string
  targets /* CommandDetailResponse_targets */ {
    devices [/*CommandDetailResponse_targets_devices*/{
      device_id  string
      status     string
      reason     string
      received_at string
    }]
    router [ /*CommandDetailResponse_targets_routers*/ {
      router_id  string
      status     string
      reason     string
      received_at string
    }]
    groups [ /* CommandDetailResponse_targets_groups*/ {
      group_id   string
      status     string
      reason     string
      received_at string
    }]
  }
}
```

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

3.8.3. 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(24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:
AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

<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
      }
    }
  ],
  "total": 1,
  "pages": 1,
  "limit": 10,
  "current_page": 1,
  "next_page": null
}
```

コマンドレスポンスの定義

```

/* CommandsResponse */
{
  commands [ /* CommandSummaryResponse*/ {
    command_id string
    url string
    name string
    accepted_at string
    status_counts /* CommandSummaryResponse_status_counts */ {
      processed integer
      rejected integer
      pending integer
    }
  }
]
total integer
pages integer
limit integer
current_page integer
next_page string
}

```

コマンド**status**一覧

コマンドレスポンスにて表示される、**status_counts**にて現在のコマンドの実行状況を表示します。

Table 3. status_counts

ステータス	説明
accepted	コマンド受理
pending	ルーターへのdownlink送信が成功し、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）、Uplinkで失敗を受信
processed	Uplinkで成功を受信

コマンド**Error**一覧

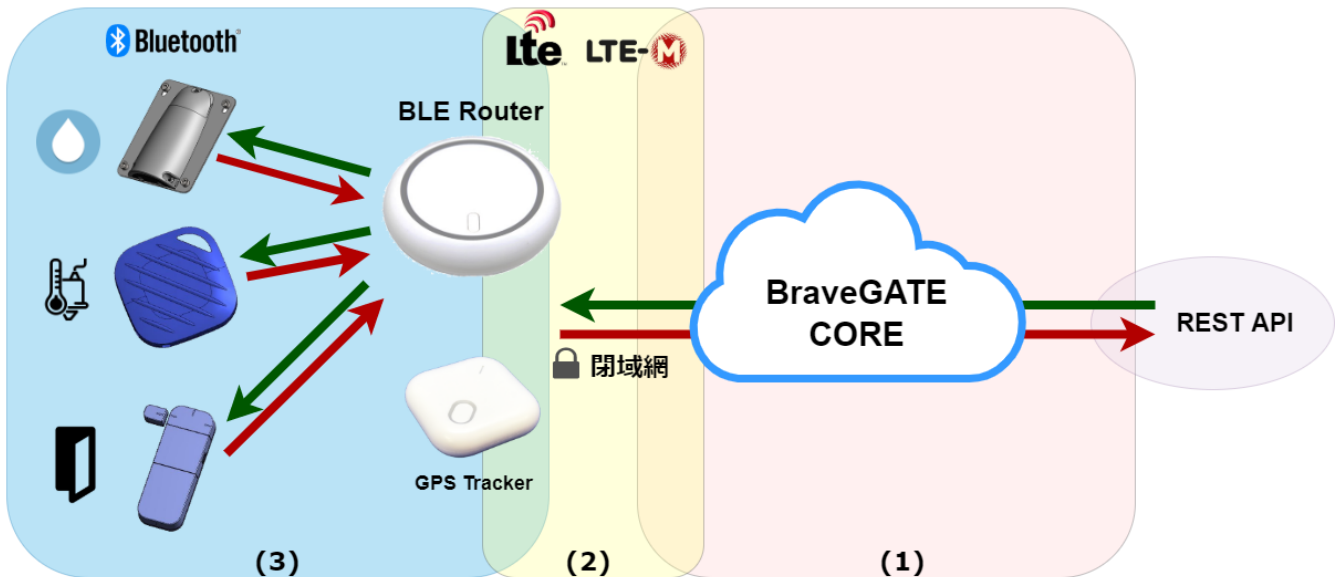
Table 4. error

エラー	説明
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 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。
sensor_id is invalid	センサーIDが不正
command not supported	対象がコマンドをサポートしていない
setting value out of range	設定値が範囲外

エラー	説明
device unreachable	デバイスへ接続できない
device timeout	デバイスとの接続がタイムアウト

3.8.4. エラーケース

各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status一覧

Table 5. (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	reason	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	reason	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に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上で定義されていない。
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 6. (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	reason	エラー内容詳細
router	router is busy.	ルーター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(ルータ側の応答がなくなる。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 7. (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	reason	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)

type	reason	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、デバイスをSCANしたけど発見できなかった。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 packets 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

```
{
  "router_id": "bb054321",
  "name": "",
  "group_id": "GRec3b8f3XXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "imsi": "440012345678901", ①
  "rssi": -68,
  "battery": 30,
  "sim_status": "ready",
  "created_at": "2019-03-25T17:24:20+09:00",
  "updated_at": "2019-03-25T17:24:20+09:00"
}
```

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-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token: TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

<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メー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
-H X-Braveridge-API-Key: AK0AqX9S3FXXXXXXXXXXXXXXXXXXXXXXXXXXXX -H X-Braveridge-Token:
TKL1b5GfhRXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

<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に割当てられている固有のID
router id	英数字8桁	routerに割当てられている固有のID
sensor id	英数字4桁	sensorの種別に割当てられている固有のID
group id	GRを接頭語とする英数字32桁	group に割当てられた固有の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 8. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	SEND_DATA_AT_ONCE
target	Object	◎	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	◎	
sensor_id	String	◎	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。

Table 9. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	SET_UPLINK_INTERVAL
target	Object	◎	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	◎	
sensor_id	String	◎	値を取得するsensor id
interval	Integer	◎	uplink間隔を秒数で指定。間隔の値はデバイスの仕様による。

例)

```
{
  "name": "SET_UPLINK_INTERVAL",
  "targets": {
    "devices": [
      "2468800001000001"
    ],
    "groups": [
      "GR0123456789abcdefghijklmnopghijklABCDEFGHIJK",
      "GR0000000000xxxxxxxxxxxxxxxxXXXXXXXXXX"
    ]
  },
  "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	値	必須	説明
name	String	◎	DOWNLOAD_DATA
target	Object	◎	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	◎	
sensor_id	String	◎	データを渡したいsensor idを指定
data	String	◎	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

(例)

```
{
  "name": "DOWNLOAD_DATA",
  "targets": {
    "devices": [
      "2468800060000001"
    ]
  },
  "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 10. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	SET_FILTER_DEVICE
target	Object	◎	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	◎	
device_id	String	◎	通信許可するdevice idを指定

例)

```

{
  "name": "SET_FILTER_DEVICE",
  "targets": {
    "routers": [
      "10000001",
      "10000002",
      "10000003"
    ],
    "groups": [
      "GR0123456789abcdefghijklABCDEFGHIJK",
      "GR0000000000xxxxxxxxxxxxxxxxXXXXXXXXXX"
    ]
  },
  "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(DOUBLE ID) (ルーターのみ)

設定されたscan対象フィルタについてフィルタを追加します。

NOTE

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 11. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	ADD_FILTER_DEVICE
target	Object	◎	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	◎	

device_id	String	◎	通信許可するdevice idを指定
-----------	--------	---	--------------------

例)

```
{
  "name": "ADD_FILTER_DEVICE",
  "targets": {
    "routers": [
      "10000001",
      "10000002",
      "10000003"
    ],
    "groups": [
      "GR0123456789abcdefghijklABCDEFGHIJK",
      "GR0000000000xxxxxxxxxxxxXXXXXXXXXXXX"
    ]
  },
  "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対象のフィルタをクリアします。Groupを指定することにより、

Table 12. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	CLEAR_FILTER_DEVICE
target	Object	◎	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

例)

```
{
  "name": "CLEAR_FILTER_DEVICE",
  "targets": {
    "routers": [
      "10000001",
      "10000002",
      "10000003"
    ],
    "groups": [
      "GR0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJK",
      "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 13. リクエストパラメーター

JSON Key	値	必須	説明
name	String	◎	POWER_OFF
target	Object	◎	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

例)


```
{
  "name": "POWER_OFF",
  "params": {
  },
  "targets": {
    "devices": [
      "2468800010000001",
      "2468800010000002",
      "2468800010000003"
    ],
    "routers": [
      "10000001",
      "10000002",
      "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 14. リクエストパラメーター

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

例) ルーター再起動

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