データサブシステム API 解説

エブリセンスジャパン株式会社

1.0 版 2017 年 01 月 26 日

データサブシステム解説

本ドキュメントは、データサブシステムについて解説します。
以下の章から成っています。

- データサブシステムの概念と構造
- API 説明書（内部 API）
- JSON over HTTP ゲートウェイとのやりとり
- センサーデータ
- 出力されるデータの形式
- センサーのリポジトリ

データサブシステムの概念と構造

データサブシステムとは

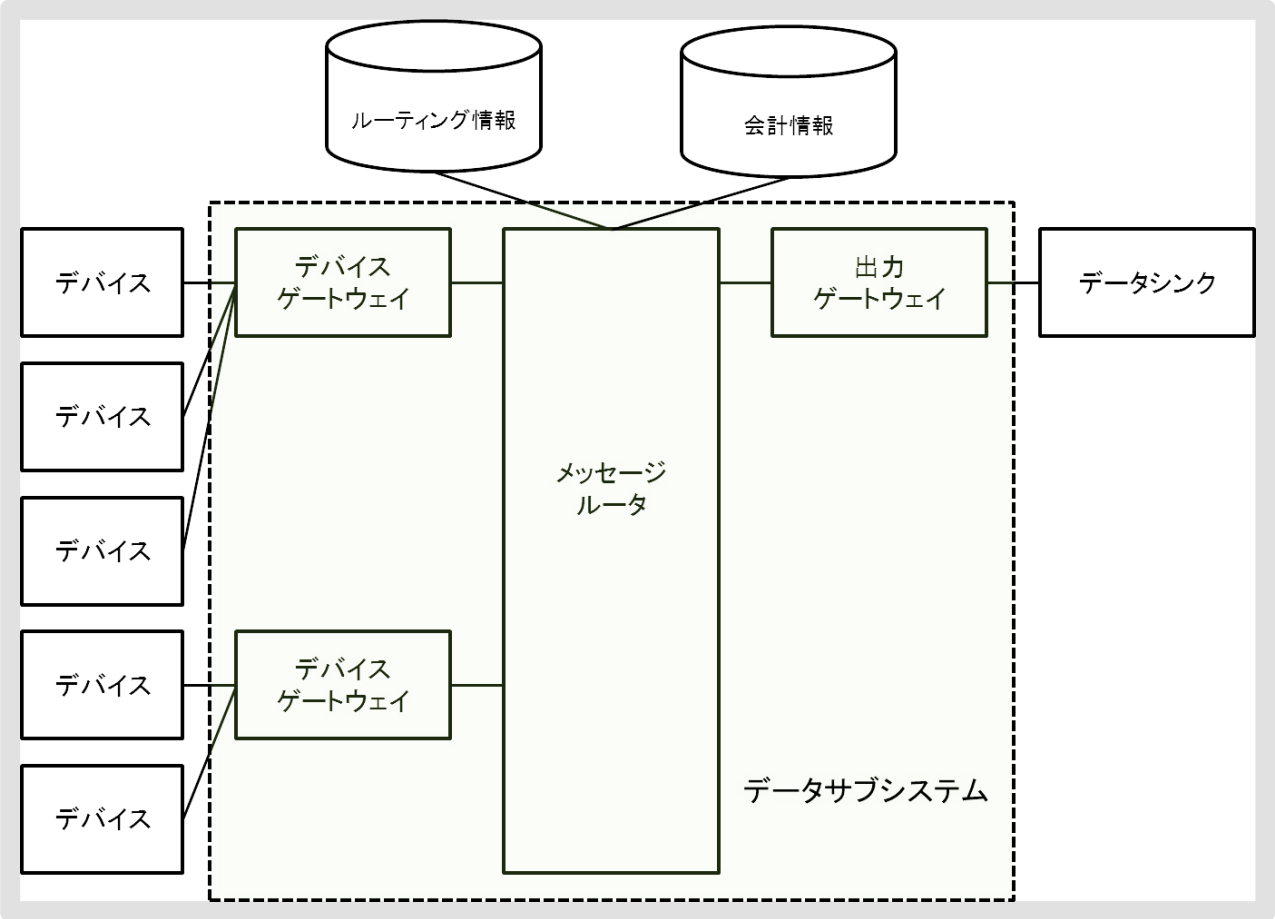
EverySense システムは、「サーバ」と「データサブシステム」から構成されています。
サーバでは、

- レシピの作成
- ファームの管理
- 会計処理

といった、センサーデータをやりとりするための、契約等の処理を主に行います。
データサブシステムは、サーバによって作られた契約情報を元に、

- データの振り分け処理
- 会計基礎データの作成
- デバイスやデータシンクとのデータ通信

といった、実際のデータのやりとりと、それによって発生する課金等の処理を行います。
データサブシステムから見た EverySense システムの構成を、以下に示します。



ファームオーナの所有するデバイスからの情報は、「デバイスゲートウェイ」を経由して「メッセージルータ」に入ります。この「メッセージルータ」がデバイスからのデータをレシピに結びつけて配送します。配送されたデータは「出力ゲートウェイ」を経由して、レストランオーナーの持っているデータシンクが受け取ります。

デバイスは様々なプロトコルでデータを送信しますが、それらのプロトコルからデータサブシステム内のプロトコルへの変換は、「デバイスゲートウェイ」が行います。現在のところ用意されているデバイスゲートウェイは、JSON over HTTP(s)に対するゲートウェイです。この他のプロトコルへの対応は、標準的なプロトコル（たとえば MQTT 等）への対応は、随時 EverySense 社の方で開発します。それ以外の特殊なプロトコルに関しては、デバイスベンダー様が開発の上の持ち込みとなります。

API 説明書（内部 API）

ここではデータサブシステムが提供する API について説明します。この API はデータサブシステムがデバイスゲートウェイに提供しているインターフェイスなので、実際にデバイスゲートウェイが外部に公開する API は、ここで挙げられている API とは異なる可能性があります。

概要

デバイスと Everysense Server との通信は、デバイスゲートウェイを通じて行われます。Everysense Server がデバイスと正しく通信するためには、デバイスゲートウェイは以下のことが出来る必要があります。

1. デバイスより受信したデータをデータサブシステムに送る
2. 受信したセンサーデータの単位系の整合
3. (必要なら)デバイスの認証
4. Everysense Server の必要に応じて、デバイスへの push

本ドキュメントでは、これらについて API の解説を行います。

前提条件

デバイスゲートウェイに提供する API の基本

デバイスゲートウェイと EverySense Server との間の通信は、全て MessagePack RPC を使います。MessagePack および MessagePack RPC についての詳しい説明については、当該ページを御覧下さい。仕様については、MessagePack specification にあります。

デバイスゲートウェイに提供する API のデータの型については、以下のものを使用します。

型名	意味
Integer	整数
Nil	nil
Boolean	論理値、 true or false
Float	浮動小数点数
String	UTF-8 文字列

Array	順序のあるオブジェクトの並び
Map	フィールド名のついたオブジェクトの並び

時刻は文字列に変換して扱われます。
API は論理的には、

```
API 名(引数 1, 引数 2,...)
```

のように呼び出されます(実際にどう表現されるかは言語に依ります)。この API 解説では

呼称	型	意味
引数 1	引数 1 の型	引数 1 の意味
引数 2	引数 2 の型	引数 2 の意味

のような形式で説明されています。
型が Map の場合、値には複数のフィールドが名前と共に列挙されます。この解説では、

名前	意味	型
name	このフィールドの意味	このフィールドの型

のように表現されます。

提供する API の種類

API は以下のようなものがあります。

- 認証 API
- セション API
- メッセージ API
- オーダーAPI
- デバイス API
- センサーAPI
- ファーム API
- ファームオーナ API
- オーナ API
- 自動承認 API
- 通知 API
- データ出力 API
- レシピ API

これらのうち、センサーデバイスで主に使うものは、

- メッセージルータ API

です。

また、レストランオーナーとしてデータを取得するのに使うものは、

- データ出力 API

です。
これ以外の API は主には EveryPost のような、Every Sense Server を操作する機能を持ったデバイスのための API です。

API の呼出方法

API は MessagePackRPC か、JSON over HTTP によって呼び出すことが出来ます。
いずれの呼び出しも、プリミティブなライブラリを組み合わせて呼び出すことが出来ますが、Ruby からの呼び出しについては OpenES が用意されています。以下はその例です。

MessagePackRPC

MessagePackRPC で API を呼び出す場合は、'every_sense_msgpack_client.rb'を使います。
たとえば、'put_message'を呼び出す場合は、

```
require 'every_sense_msgpack_client'
server = EverySense::MsgPackClient.new('api.every-sense.com')
p server.put_message("fc1bfbd8-2b20-473e-8cd6-6030b2402ec1",
                     [
                      {
                        "data" => {
                          "at" => Time.now.to_s,
                          "unit" => "degree",
                          "location" => ["35.705570","139.770767"],
                          "elevation" => "0.000000",
                          "datum" => "WGS84",
                          "memo" => ""
                        },
                        "sensor_name" => "GPS"
                      }])
```

のように行います。

JSON

API を JSON から呼ぶ場合は、

```
http://<ホスト名>:<ポート番号>/<メソッド名>
```

の形式の URL に POST します（一部の参照系の API では GET もあります）。

- ホスト名
  現在は'api.every-sense.com'です。
- ポート番号
  現在は 7001 です。8001 を使うと https でアクセスされます。
- メソッド名
  API のメソッド名そのものです。
- POST の内容
  JSON の配列であり、要素の順序は API の引数の順序に一致させます。

たとえば、'resolv'を呼び出す場合は、

```
http://<ホスト名>:<ポート番号>/resolv
```

に

```
[ <デバイス UUID> , <ローカル名> ]
```

を POST します。

Ruby から呼び出す場合は、'every_sense_json_client.rb'を使います。
たとえば、'put_message'を呼び出す場合は、

```
require 'every_sense_json_client'
server = EverySense::JsonClient.new
p server.put_message("fc1bfbd8-2b20-473e-8cd6-6030b2402ec1",
                     [
                      {
                        "data" => {
                          "at" => Time.now.to_s,
                          "unit" => "degree",
                          "location" => ["35.705570","139.770767"],
                          "elevation" => "0.000000",
                          "datum" => "WGS84",
                          "memo" => ""
                        },
                        "sensor_name" => "GPS"
                      }])
```

のように行います。

API の認証

プライベートなデータの参照およびデータの更新を行う API には、認証を行うものがあります。

返り値

多くの API では、以下のフィールドを含む *Map* 形式で結果が返ります。これ以外の結果を返すものもあります。
特にことわりのない場合、API の説明は以下のフィールドについての説明を省略します。

名前	型	意味
code	Integer	復帰コード
reason	Text	エラー説明（復帰コードが-1, -2 の場合）
message	Text	エラーメッセージ（復帰コードが-10, -20 の場合）
trace	Text	エラートレース（復帰コードが-20 の場合）

復帰コード

- 0: 成功
- -1: 失敗あるいはデータなし
- -2: 認証エラー
- -10:デバイスゲートウェイ内でのエラー(通常パラメータ指定の間違い)
- -20: データサブシステムのバグによるエラー

認証 API

概要

認証 API は、利用者情報の登録変更と、認証を行うためのものです。

API

認証情報についての API には、以下のものがあります。

- new_user
  新規ユーザの登録
- auth_user
  認証
- get_user
  ユーザ情報の取得
- update_user
  ユーザ情報の変更

new_user

概要

新しいユーザを作ります。

説明

new_user は、ユーザ情報を与えて新規のユーザを作ります。

- String ログイン ID
  ログインに使う ID
- String 表示名
  ログイン用の ID とは別に画面上に表示する任意の名前
- String メールアドレス
- String パスワード
  ログインパスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - String ユーザ UUID(uuid)
    new_user が成功すると、ユーザの uuid が返ります。失敗した場合は何も返りません

auth_user

概要
ユーザを承認します。

説明
auth_user は、ログイン ID とパスワードを与えて、認証を行います。

- String ログイン ID
  ユーザのログイン ID です
- String パスワード
  ユーザのパスワードです

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - String ユーザ UUID(uuid)
    auth_user が成功すると、ユーザの uuid が返ります。失敗した場合は何も返りません

get_user

概要
ユーザ情報を取得します。

説明
get_user は、ユーザ UUID を与えて、ユーザ情報を取得します。

- String ユーザ UUID
  ユーザに付与された UUID
- String パスワード
  パスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -2: 認証エラー
  - String ユーザ UUID(uuid)
  - String 表示名(name)
  - String メールアドレス(mail)

失敗した場合は、uuid, name, mail は返りません。

update_user

概要
ユーザ情報を変更します。

説明
update_user は、ユーザ情報を与えて、ユーザ情報を更新します。

- String ユーザ UUID
  ユーザに付与された UUID
- String パスワード
  現在のパスワードです
- Map オプション
  更新する項目です。以下のものが更新可能です
    - String 表示名(name)
    - String メールアドレス(mail)
    - String 変更後のパスワード(password)

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー

セション API

概要

セション API は、認証済みのセションを管理する API です。
セションは認証によって作成され、作成後一定期間(現在は 1 週間)有効です。
セションが使用されると、使用された時から一定期間(現在は 1 週間)、有効期間が延長されます。
セションの正当性は、セションキーの検証によってのみ行われます。セションキーが第三者に漏洩した場合は、すみやかに無効化する必要があります。
現在のところ、セションでの認証が有効な API は、データ出力 API のみです。

API

セションについての API は、以下のものがあります。

- create_session
  認証用のセションを作ります
- check_session
  セションが有効かどうか調べます
- delete_session
  セションを無効にします

create_session

概要

セションを作成します。

説明

create_session はセションを作成し、セションキーを返します。

- String ユーザ UUID
- String パスワード
- String セションの説明(省略可)

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -2: 認証エラー
  - String セションキー(session_key)

成功しなかった場合は session_key は返りません。

check_session

概要
セションが有効かどうか調べます。

説明
check_session は、与えられたセションキーが有効かどうか調べます。

- String セションキー
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: セションなし

delete_session

概要
セションを削除します。

説明
delete_session は与えられたセションキーを無効化します。

- String ユーザ UUID
- String パスワード
- String セションキー

返り値
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: セションなし
      -2: 認証エラー

メッセージ API

概要

ファームで収集したセンサーデータを、レシピに従いレストランに配送します。
デバイスゲートウェイから見た場合、デバイスゲートウェイが受けつけたデータを吐き出す先になります。

API

デバイスゲートウェイからメッセージルータに対する API の機能(function)は、以下のものがあります。

- put_message
  受信したデータをメッセージルータに送ります。
- resolv
  センサーローカル名からセンサーUUID を求めます。
- get_message
  入力キューにある受信データを取得します。

put_message

概要

受信したデータを、メッセージルータに送ります。

説明

put_message は、デバイスゲートウェイからのデータを、メッセージルータに送ります。

呼称	型	意味
デバイス UUID	String	デバイスに付与された uuid
データの並び	Array of Object	ファームから収集されたデータ

個々のセンサーデータは、以下のようになります。

名前	呼称	型

sensor_uuid	センサーUUID	String
sensor_name	センサーローカル名	String
data	センサーデータ	Object

「センサーローカル名」または「センサーUUID」で識別されます。「センサーローカル名」と「センサーUUID」はいずれかを指定します。

センサーデータの具体的な内容については、センサーデータにあります。

返り値

標準の返り値が返ります。

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗

注意

デバイス UUID が正当な場合は、見掛け上処理は成功します。ただし、**実際にデータが正しく配送されるかどうかは、データの内容やレシピの状態に依存**します。

resolv

概要

センサーローカル名からセンサーUUID を求めます。

説明

resolv は、デバイスとセンサー名からセンサーUUID を求めます。

呼称	型	意味
デバイス UUID	String	デバイスに付与された uuid
センサーローカル名	String	デバイス上のセンサーにつけられた名前

返り値

返り値は以下のフィールドを持つ MAP です。この他に標準のフィールドを持ちます。

名前	型	意味
uuid	String	センサーUUID

get_message

概要

入力キューにある受信データを取得します。

説明

get_message は、メッセージルータの入力キューにある受信データを取得します。

呼称	型	意味
ユーザ uuid	String	ユーザの UUID
パスワード	String	
デバイス UUID	String	データを取得するデバイスの UUID

返り値

以下のフィールドを持つ Map です。この他に標準のフィールドを持ちます。

名前	型	意味
count	Integer	データ件数
inputs	Object Array	データ

出力されるデータの形式については、出力されるデータの形式を参照して下さい。

オーダーAPI

概要

オーダーAPI は、ファームへの「注文」の読み書きを行います。

API

オーダーAPI は以下のものがあります。

- get_order_list
  デバイスに関係のあるオーダーの一覧を得ます
- get_order_list2
  デバイスに関係のあるオーダーの一覧を得ます
- get_order
  オーダーの内容を得ます
- get_order_list_with_entry
  デバイスに関係のあるオーダーの内容の一覧を得ます
- get_order_list_with_entry_only_header
  デバイスに関係のあるオーダーの内容（一部）の一覧を得ます
- update_order
  オーダーの状態を変更します
- update_orders
  オーダーの状態を変更します(まとめて)
- delete_order
  オーダーを削除します
- get_real_requests
  稼動状態を取得します
- update_order_shared_secret
  オーダーのシェアードシークレットを変更します

get_order_list

概要

デバイスに関係のあるオーダーの一覧を得ます。

説明

get_order_list は、デバイス UUID を指定して、指定期間内にあるオーダーを得ます。

- String デバイス UUID
  デバイスの付与された UUID です

- String 取得対象
  得ようとするオーダーの期間を指定します。以下のような形式です
  - 'YYYYMMDDhhmmss-':指定時刻以後に更新されたもの
  - '-YYMMDDhhmmss': 指定時刻以前に更新されたもの
  - 'YYMMDDhhmmss-YYMMDDhhmmss': 指定範囲に更新されたもの

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - Array オーダーUUID の配列(uuids)
    - String オーダーの UUID

get_order_list2

概要

デバイスに関係のあるオーダーの一覧を得ます。

説明

get_order_list2 は、デバイス UUID を指定して、指定期間内にあるオーダーを得ます。

- String デバイス UUID
  デバイスの付与された UUID です
- String 開始日時
  得ようとするオーダーの期間の開始を指定します。
- String 終了日時
  得ようとするオーダーの期間の終了を指定します。

日時については、以下のように解釈されます。

- 開始日時、終了日時共に指定された場合
  指定範囲に更新されたもの
- 開始日時のみ指定された場合
  指定時刻以後に更新されたもの
- 終了日時のみ指定された場合
  指定時刻以前に更新されたもの

- いずれも指定されていない場合
  デバイスに対するオーダーの全てが返ります

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - Array オーダーUUID の配列(uuids)
    - String オーダーの UUID

get_order

オーダーの内容を得ます。

説明

get_order はオーダーUUID を指定して、オーダーの内容を得ます。

- String オーダーUUID
  オーダーに付与された UUID

返り値

成功すると、オーダーの内容が Map で返ります。
オーダーが取得出来ない場合は nil が返ります。

オーダーの内容は以下のようになります。

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - String オーダー発行元名(organization)
  - Integer レストランオーナーの評価(restaurant_owner_rank)
  - String レストランオーナーWEBURL(restaurant_owner_web_url)
  - String レストランオーナー業種(restaurant_owner_type_of_business)
  - String レシピ名(recipe)
  - String 説明(description)
  - String 利用目的(purpose)
  - String ファーム UUID(farm_uuid)

- String ファームクラス名(farm_class)
- Array センサーUUID(sensor_uuids)
    - Map センサー
        - String センサーUUID
        - String 出力単位(output_level)
- String 募集開始日時(application_period_start)
- String 募集終了日時(application_period_end)
- String データ収集開始日時(collection_period_start)
- String データ収集終了日時(collection_period_end)
- String イベント名(event_trigger)
- Integer 収集間隔(time_interval)
- String 収集間隔の単位(time_unit)
- String 計測データ最大受信遅延時間(time_maximum_latency)
- String 計測データ最大受信遅延時間の単位(time_maximum_latency_unit)
- String 計測時刻の許容範囲(time_permissible_range)
- String 計測時刻の許容範囲の単位(time_permissible_range_unit)
- String ファームの状態(farm_status) none, executable, suspend
- String 更新日時(updated_at)
- String 削除日時(deleted_at)
- Boolean 第三者提供(third_party)
- String 第三者提供：提供先(third_party_name)
- Boolean 商用利用(commercial_use)
- Boolean 条件不一致フラグ(mismatched)
- Boolean オーダー無効(order_disabled)
  センサーが無効もしくは、ファームからセンサーが削除された場合に true を返します。
- String シェアードシークレット(shared_secret)
- String 審査(judge_type)
  none: 審査なし
  judge: 審査あり
- String 審査状態(judge_status)
  unuse: 不使用
  use: 使用
  judge_type が judge の場合にのみ judge_status の状態に意味を持ちます。
- Boolean ポイント提供あり(point_supply_enabled)
- String ポイント率(point_rate)
- Integer 最低ポスト回数(minimum_post_count_to_point)
- Integer ボーナスポイント(bonus_point)
- Integer 完了ポスト回数(minimum_post_count_to_complete)
- Integer 現在のポスト回数(now_post_count)

- String 最大獲得ポイント(maxmum_gain_point)
- String 最大ポスト数(maximum_posts)

失敗した場合、復帰コード以外のものは返りません。

get_order_list_with_entry

概要

デバイスに関係のあるオーダーの内容の一覧を得ます。

説明

get_order_list_with_entry は、デバイス UUID を指定して、指定期間内にあるオーダーの内容の一覧を得ます。すなわち、get_order_list と get_order とを併せた機能を持ちます。

- String デバイス UUID
  デバイスの付与された UUID です
- String 取得対象
  得ようとするオーダーの期間を指定します。以下のような形式です
  - 'YYYYMMDDhhmmss-':指定時刻以後に更新されたもの
  - '-YYMMDDhhmmss': 指定時刻以前に更新されたもの
  - 'YYMMDDhhmmss-YYMMDDhhmmss': 指定範囲に更新されたもの

返り値

成功すると、オーダーの内容が Map で返ります。
オーダーが取得出来ない場合は nil が返ります。

オーダーの内容は以下のようになります。

Map 返り値

- Integer 復帰コード(code)
  0: 成功
  -1: 失敗
- Array オーダーの配列(orders)
  - Map オーダー
    - String オーダー発行元名(organization)
    - Integer レストランオーナーの評価(restaurant_owner_rank)
    - String レストランオーナーWEBURL(restaurant_owner_web_url)
    - String レストランオーナー業種(restaurant_owner_type_of_business)
    - String レシピ名(recipe)
    - String 説明(description)

- String 利用目的(purpose)
- String ファーム UUID(farm_uuid)
- String ファームクラス名(farm_class)
- Array センサーUUID(sensor_uuids)
    - Map センサー
        - String センサーUUID
        - String 出力単位(output_level)
- String 募集開始日時(application_period_start)
- String 募集終了日時(application_period_end)
- String データ収集開始日時(collection_period_start)
- String データ収集終了日時(collection_period_end)
- String イベント名(event_trigger)
- Integer 収集間隔(time_interval)
- String 収集間隔の単位(time_unit)
- String 計測データ最大受信遅延時間(time_maximum_latency)
- String 計測データ最大受信遅延時間の単位(time_maximum_latency_unit)
- String 計測時刻の許容範囲(time_permissible_range)
- String 計測時刻の許容範囲の単位(time_permissible_range_unit)
- String ファームの状態(farm_status)
- String 更新日時(updated_at)
- String 削除日時(deleted_at)
- Boolean 第三者提供(third_party)
- String 第三者提供：提供先(third_party_name)
- Boolean 商用利用(commercial_use)
- Boolean 条件不一致フラグ(mismatched)
- Boolean オーダー無効(order_disabled)
  センサーが無効もしくは、ファームからセンサーが削除された場合に true を返します。
- String シェアードシークレット(shared_secret)
- String 審査(judge_type)
  none: 審査なし
  judge: 審査あり
- String 審査状態(judge_status)
  unuse: 不使用
  use: 使用
  judge_type が judge の場合にのみ judge_status の状態に意味を持ちます。
- Boolean ポイント提供あり(point_supply_enabled)
- String ポイント率(point_rate)
- Integer 最低ポスト回数(minimum_post_count_to_point)
- Integer ボーナスポイント(bonus_point)

        - Integer 完了ポスト回数(minimum_post_count_to_complete)
        - Integer 現在のポスト回数(now_post_count)
        - Integer ポスト率(order_score_post_rate)
        - String 最大獲得ポイント(maxmum_gain_point)
        - String 最大ポスト数(maximum_posts)

get_order_list_with_entry_only_header

概要

デバイスに関係のあるオーダーの内容（一部）の一覧を得ます。

説明

get_order_list_with_entry は、デバイス UUID を指定して、指定期間内にあるオーダーの内容（一部）の一覧を得ます。すなわち、get_order_list と get_order とを併せた機能を持ちます。

- String デバイス UUID
  デバイスの付与された UUID です
- String 取得対象
  得ようとするオーダーの期間を指定します。以下のような形式です
  - 'YYYYMMDDhhmmss-':指定時刻以後に更新されたもの
  - '-YYMMDDhhmmss': 指定時刻以前に更新されたもの
  - 'YYMMDDhhmmss-YYMMDDhhmmss': 指定範囲に更新されたもの

返り値

成功すると、オーダーの内容が Map で返ります。
オーダーが取得出来ない場合は nil が返ります。

オーダーの内容は以下のようになります

Map 返り値

- Integer 復帰コード(code)
  0: 成功
  -1: 失敗
- Array オーダーの配列(orders)
    - Map オーダー
        - String レシピ名(recipe)
        - Array センサーUUID(sensor_uuids)
            - Map センサー
                - String センサーUUID(uuid)

        - String 募集開始日時(application_period_start)
        - String 募集終了日時(application_period_end)
        - String データ収集開始日時(collection_period_start)
        - String データ収集終了日時(collection_period_end)
        - Integer 収集間隔(time_interval)
        - String 収集間隔の単位(time_unit)
        - String ファームの状態(farm_status)
        - String 更新日時(updated_at)
        - String 削除日時(deleted_at)
        - Boolean 条件不一致フラグ(mismatched)
        - Boolean オーダー無効(order_disabled)
          センサーが無効もしくは、ファームからセンサーが削除された場合に true を返します。
        - Boolean ポイント提供あり(point_supply_enabled)
        - String 最大獲得ポイント(maxmum_gain_point)

update_order

オーダーの状態を変更します。

説明

update_order は、オーダーの状態を変更します。

- String ユーザ UUID
- String パスワード
- String オーダーUUID
- String ステータス
  none, allow, wait_for_active, active, done, deny, cancel, suspend
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー

update_orders

オーダーの状態を変更します(まとめて)

説明

update_order は、オーダーの状態を複数まとめて変更します。

- String ユーザ UUID
- String パスワード
- Array オーダーUUID と状態の配列
    - Array
        - String オーダーUUID
        - String ステータス
          none, allow, wait_for_active, active, done, deny, cancel, suspend

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - Integer 処理件数(count)
      処理に成功した件数

delete_order

オーダーを削除します。

説明

delete_order は、オーダーを削除します。

- String ユーザ UUID
- String パスワード
- String オーダーUUID

または

- Array
    - String オーダーUUID
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - Integer 処理件数(count)

get_real_requests

稼動状態を取得します。

説明

get_real_requests は、オーダーの稼動状態を取得します。

- String ユーザ UUID
- String パスワード
- String オーダーUUID
- String 取得開始日時(デフォルトはデータ取得開始日時)
- String 取得終了日時(デフォルトは今)

※取得開始日時、取得終了日時ともに指定しない場合は、
・取得終了日時には、今もしくは今がレシピのデータ取得期間終了日時を超える場合は、レシピのデータ取得期間終了日時
・取得開始日時には、取得終了日時から 10 日さかのぼった日時、取得終了日時から 10 日遡った日時がレシピのデータ取得期間開始日時より前になる場合はレシピのデータ取得期間開始日時
が設定されます。

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - Array 稼動状態(results)
        - Array
            - String 日
              "YYYY mm/dd" というフォーマットです
            - Number ポスト数

update_order_shared_secret

オーダーのシェアードシークレットを変更します。

説明

update_order_shared_secret は、オーダーのシェアードシークレットを変更します。

- String ユーザ UUID

- String パスワード
- String オーダーUUID
- String シェアードシークレット

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー

デバイス API

概要

デバイス API はデバイス情報の操作を行います。

API

デバイス API は以下のものがあります。

- get_device
  利用可能なデバイス情報を返します。
- create_device
  デバイスを作成します。
- get_device_info
  指定したデバイス情報とセンサー情報を返します。
- update_device_info
  指定したデバイス情報とセンサー情報を更新ます。
- add_device_token
  指定したデバイス情報に通知用のトークンを登録します。
- get_device_class
  指定したデバイスクラスの情報を返します。

get_device

利用可能なデバイス情報を返します。

説明

get_device は、デバイスクラスとユーザ UUID を指定して、そのユーザで有効のデバイス UUID とデバイス名称を返します。

- String ユーザ UUID
- String パスワード
- String デバイスクラス名
  デバイスクラスを指定します

返り値

- Map 返り値

    - Integer 復帰コード(code)
      0: 存在する
      -1: 存在しない
      -2: 認証エラー
    - Array デバイスの配列(devices)
        - Map デバイス
            - String デバイス UUID(uuid)
            - String デバイス名称(name)

create_device

デバイスを登録します。

説明

create_device_entry は、必要な情報を与えてデバイスを登録します。
デバイスクラスは最新バージョンが適用されます。

- String ユーザ UUID
- String パスワード
- String ドライバー名
  作成したいデバイスクラスのドライバー名を指定します
- String デバイス名
  デバイスに付ける名前を指定します
- Boolean 有効
  デバイスを有効にするかどうかを指定します

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - String デバイス UUID(uuid)

成功しなかった場合は uuid は返りません。

get_device_info

指定したデバイス情報とセンサー情報を返します。

説明

get_device_info は、ユーザ UUID とデバイス UUID から、そのデバイス情報とセンサー情報を返します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID
  デバイス UUID を指定します

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 存在する
    -1: 存在しない
    -2: 認証エラー
  - String デバイス UUID(uuid)
  - String デバイス名称(name)
  - Boolean 有効(enabled)
  - String デバイスクラス名(device_class_name)
  - String ドライバー名(driver_name)
  - String バージョン(version)
  - Array センサーの配列(sensor_list)
    - Map センサー
      - String センサーUUID(uuid)
      - String センサー名(name)
      - String センサークラス名(class_name)
      - String 表示名称(display_name)
      - String 説明(description)
      - Boolean 有効(enabled)
      - String 設置場所種別(location_type)
      - String 設置位置：郵便番号(location_point_zipcode)
      - String 設置位置：都道府県(location_point_prefectures)
      - String 設置位置：市区町村(location_point_city)
      - String 設置位置：アドレス(location_point_address)
      - String 設置場所(location_in_out)
      - String 設置場所詳細(location_detail)
      - Map 精度(accuracy)
        - String 精度種別(type)
        - Integer 値(value)

        - Integer 最小目盛(minimum_scale)
        - Map レンジ(range)
            - Integer 最小計測範囲(min)
            - Integer 最大計測範囲(max)
        - Map 停止時刻(receive_window_time)
            - String 停止開始時刻(close_time)
            - String 停止終了時刻(open_time)
        - String 標準単位(default_unit)

パラメータの詳細

- 設置場所(location_in_out)
  undefined 未設定
  indoor 屋内
  outdoor 屋外
- 設置場所種別(location_type)
  undefined 未設定
  fix 固定
  movable 移動

update_device_info

指定したデバイス情報とセンサー情報を更新ます。

説明

update_device_info は、ユーザ UUID とデバイス UUID から、そのデバイス情報とセンサー情報を返します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID
  デバイス UUID を指定します
- Stiring デバイス名
- Boolean 有効
- Array センサーの配列
    - Map センサー
        - String センサーUUID(uuid)
        - Boolean 有効(enabled)
        - String 設置場所種別(location_type)

        - String 設置位置：都道府県(location_point_prefectures)
        - String 設置位置：市区町村(location_point_city)
        - String 設置位置：アドレス(location_point_address)
        - String 設置場所(location_in_out)
        - String 設置場所詳細(location_detail)

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 存在する
      -1: 存在しない
      -2: 認証エラー
    - String デバイス uuid(uuid)

備考

設置場所種別が「固定」以外の場合、「設置位置：都道府県」「設置位置：市区町村」「設置位置：アドレス」「設置位置：設置場所」「設置場所詳細」で指定された値は無視されます。

add_device_token

指定したデバイス情報に通知用のトークンを登録します。

説明

add_device_token は、ユーザ UUID とデバイス UUID とトークンを受け取り、そのデバイスにトークンを登録します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID
  デバイス UUID を指定します
- Stiring トークン

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 存在する
      -1: 存在しない
      -2: 認証エラー

    - String デバイス uuid(uuid)

get_device_class

指定したデバイスクラスの情報を返します。

説明

get_device_class は、デバイスクラスのドライバー名を渡すことで、デバイスクラスの情報を返します。

- String ドライバー名
- String バージョン
  省略すると、最新バージョンのデバイスクラスの情報を取得します。

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 存在する
      -1: 存在しない
    - String デバイスクラス名(name)
    - String バージョン(version)
    - String 説明(description)

センサーAPI

概要

センサーAPI は、センサーの情報を操作します。

API

センサーAPI には以下のものがあります。

- get_sensor
  センサーの情報を得ます
- get_sensors
  センサーの情報を得ます(複数指定)
- update_sensor
  センサーの情報を更新します

get_sensor

センサーの情報を得ます。

説明

get_sensor は、指定されたセンサーの内容を得ます。

- String センサーUUID

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - String センサーUUID(uuid)
  - String センサー名(name)
    デバイスではセンサー名（DeviceDriver 上で一意になる文字列。任意で設定します）で実際のセンサーがわかるようにします。
    これは DeviceClass でも定義します。
  - String センサークラス名(class_name)
  - String 表示名称(display_name)
  - String 説明(description)
  - Boolean 有効(enabled)

        - String 設置場所種別(location_type)
          undefined 未設定/fix 固定/movable 移動
        - String 設置位置：都道府県(location_point_prefectures)
        - String 設置位置：市区町村(location_point_city)
        - String 設置位置：アドレス(location_point_address)
        - String 設置場所(location_in_out)
          undefined 未設定/indoor 屋内/outdoor 屋外
        - String 設置場所詳細(location_detail)
            - Map 精度(accuracy)
                - String 精度種別(type)
                - Integer 値(value)
            - Integer 最小目盛(minimum_scale)
            - Map レンジ(range)
                - Integer 最小計測範囲(min)
                - Integer 最大計測範囲(max)
            - Map 停止時刻(receive_window_time)
                - String 停止開始時刻(close_time)
                - String 停止終了時刻(open_time)
            - String 標準単位(default_unit)

成功しなかった場合は、復帰コード以外はありません。

get_sensors

センサーの情報を得ます。

説明

get_sensors は、指定されたセンサーの内容を得ます。
複数同時に指定できます。

- Array センサーの配列
    - String センサーUUID

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
    - String 失敗理由(reason)
    - Integer 取得成功した個数(count)

- Array センサー情報の配列(sensors)
  - String センサーUUID(uuid)
  - String センサー名(name)
    デバイスではセンサー名（DeviceDriver 上で一意になる文字列。任意で設定します）で実際のセンサーがわかるようにします。
    これは DeviceClass でも定義します。
  - String センサークラス名(class_name)
  - String 表示名称(display_name)
  - String 説明(description)
  - Boolean 有効(enabled)
  - String 設置場所種別(location_type)
    undefined 未設定/fix 固定/movable 移動
  - String 設置位置：都道府県(location_point_prefectures)
  - String 設置位置：市区町村(location_point_city)
  - String 設置位置：アドレス(location_point_address)
  - String 設置場所(location_in_out)
    undefined 未設定/indoor 屋内/outdoor 屋外
  - String 設置場所詳細(location_detail)
    - Map 精度(accuracy)
      - String 精度種別(type)
      - Integer 値(value)
    - Integer 最小目盛(minimum_scale)
    - Map レンジ(range)
      - Integer 最小計測範囲(min)
      - Integer 最大計測範囲(max)
    - Map 停止時刻(receive_window_time)
      - String 停止開始時刻(close_time)
      - String 停止終了時刻(open_time)
    - String 標準単位(default_unit)

成功しなかった場合は、復帰コードと失敗理由以外はありません。
成功した場合は、復帰コードと成功した個数と情報が返ります。

update_sensor

センサーの情報を更新します。

説明

update_sensor は、指定されたセンサーの内容を更新します。

- String ユーザ UUID
- String パスワード
- String センサーUUID(uuid)
- Boolean 有効(enabled)
- String 設置場所種別(location_type)
  undefined 未設定/fix 固定/movable 移動
- String 設置位置：都道府県(location_point_prefectures)
- String 設置位置：市区町村(location_point_city)
- String 設置位置：アドレス(location_point_address)
- String 設置場所(location_in_out)
  undefined 未設定/indoor 屋内/outdoor 屋外
- String 設置場所詳細(location_detail)
- String 停止開始時刻(receive_window_close_time)
  タイムゾーン UTC の時刻で形式は"HH:mm"で指定してください。
- String 停止終了時刻(receive_window_open_time)
  タイムゾーン UTC の時刻で形式は"HH:mm"で指定してください。

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー

備考

設置場所種別が「固定」以外の場合、「設置位置：都道府県」「設置位置：市区町村」「設置位置：アドレス」「設置位置：設置場所」「設置場所詳細」で指定された値は無視されます。

ファーム API

概要

ファーム API はファームの操作を提供します。

API

ファーム API には以下のものがあります。

- get_farm_list
  デバイスにあるファームの一覧を得ます
- get_farm
  ファームの内容を得ます
- get_farm_list_with_entry
  デバイスにあるファームの内容を全て返します
- create_farm
  ファームを作成します
- update_farm
  ファームを更新します

get_farm_list

概要

デバイスにあるファームの一覧を得ます。

説明

get_farm_list は、デバイス UUID を指定してファーム UUID のリストを得ます。

- String ユーザ UUID
- String パスワード
- String デバイス UUID

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー
  - Array ファーム UUID の配列(uuids)
    - String ファーム UUID

get_farm

ファームの内容を得ます。

説明

get_farm は、ファーム UUID を指定してファームの内容を得ます。

- String ユーザ UUID
- String パスワード
- String ファーム UUID

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - String ファーム UUID(uuid)
    - String ファーム名(name)
    - Boolean 有効(enabled)
    - Array センサー情報の配列(sensors)
        - String センサーUUID(uuid)
        - String センサー名(name)
        - String 表示名称(display_name)
        - String 説明(description)
        - Boolean 有効(enabled)

get_farm_list_with_entry

デバイスにあるファームの内容を全て返します。

説明

get_farm_list_with_entry は、デバイスにあるファームの一覧を、内容と共に返します。すなわち、get_farm_list と get_farm を併せた動作をします。

- String ユーザ UUID
- String パスワード
- String デバイス UUID

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功

-1: 失敗
-2: 認証エラー
- Array ファーム情報の配列(farms)
  - String ファーム UUID(uuid)
  - String ファーム名(name)
  - Boolean 有効(enabled)
  - Array センサー情報の配列(sensors)
    - String センサーUUID(uuid)
    - String 有効(enabled)
    - String センサー名(name)
    - String 表示名称(display_name)
    - String 説明(description)

create_farm

ファームを作成します。

説明

create_farm はファームを作成します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID
- String ファーム名
- Boolen 有効

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー
  - String ファーム UUID(uuid)

update_farm

ファームを更新します。

説明

update_farm はファームを更新します。

- String ユーザ UUID
- String パスワード
- String ファーム UUID

- String ファーム名
- Boolen 有効

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー

ファームオーナ API

概要

ファームオーナ API は、ファームオーナの操作を行います。

API

ファームオーナ API には以下のものがあります。

- get_farm_owner
  ファームオーナの情報を得ます
- create_farm_owner
  ファームオーナを作成します
- update_farm_owner
  ファームオーナの情報を更新します
- get_farm_owner_fields
  ファームオーナフィールドの一覧を得ます

get_farm_owner

ファームオーナの情報を得ます。

説明

get_farm_owner は、ファームオーナに関する情報を取得します。

- String ユーザ UUID
- String パスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー
  - Array オーナフィールドの配列(owner_fields)
    - Map オーナフィールド
      - String フィールド名(name)
      - String フィールドの型(type)
        LIST, RADIO, CHECKBOX, STRING, INTEGER, DATE, LOCATION
      - Integer または String フィールドの値
      - Array 選択肢
        フィールドの型が選択肢を持つ場合に存在する
        - String 選択肢のラベル
        - String 選択肢の値

    - String オーナ UUID(uuid)

注意

自分の情報しか得ることが出来ません。

create_farm_owner

ファームオーナを作成します。

説明

create_farm_owner は、与えられた情報からファームオーナを作成します。

- String ユーザ UUID
- String パスワード
- Array ファームオーナフィールドの配列
    - Map ファームオーナフィールド
        - String 名前(name)
        - String 値(value)

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - String ファームオーナ UUID(uuid)

成功しなかった場合は、復帰コード以外は返りません。

注意

自分にしか作ることが出来ません。

update_farm_owner

ファームオーナの情報を更新します。

説明

update_farm_owner は、ファームオーナの情報を更新します。

- String ユーザ UUID
- String パスワード
- Array ファームオーナフィールドの配列

    - Map ファームオーナフィールド
        - String 名前(name)
        - String 値(value)

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -2: 認証エラー
    - Integer 更新に成功したフィールドの数(count)

count は成功した場合のみ返ります。

注意

自分のものしか更新できません。

get_farm_owner_fields

ファームオーナフィールドの一覧を得ます。

説明

get_farm_owner_fields は、ファームオーナフィールドの一覧を得ます。
引数はありません。

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
    - Array ファームオーナフィールドの配列(fields)
    - Map ファームフィールド
        - String フィールド名(name)
        - String フィールドの型(type)
          LIST, RADIO, CHECKBOX, STRING, INTEGER, DATE, LOCATION
        - Integer または String フィールドの値
        - Array 選択肢
          フィールドの型が選択肢を持つ場合に存在する
            - String 選択肢のラベル
            - String 選択肢の値

注意

失敗することはありません。

オーナ API

概要

オーナ API は、オーナの操作を行います。

API

オーナ API には以下のものがあります。

- get_owner
  オーナの情報を得ます
- create_owner
  オーナを作成します
- update_owner
  オーナの情報を更新します
- get_owner_fields
  オーナのフィールド一覧を得ます
- get_prefectures
  入力可能な住所のリストを得ます
- get_owner_id
  オーナ id を得ます

get_owner

オーナの情報を得ます。

説明

- String ユーザ UUID
- String パスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー
  - Array オーナフィールドの配列(owner_fields)
    - Map オーナフィールド
      - String フィールド名(name)
      - String フィールドの型(type)
        LIST, RADIO, CHECKBOX, STRING, INTEGER, DATE, LOCATION

        - Integer または String フィールドの値
        - 公開・非公開設定(private)
        - Array 選択肢
          フィールドの型が選択肢を持つ場合に存在する
            - String 選択肢のラベル
            - String 選択肢の値
        - 補助フィールド(variable_name)
          住所の項目にのみ付加
        - Boolean 必須項目かどうか(required)
    - String オーナ UUID(uuid)

失敗した場合は、復帰コード以外は返りません。

注意
自分の情報しか得られません。

create_owner

create_owner は、与えられた情報からオーナを作成します。

説明
- String ユーザ UUID
- String パスワード
- Array ファームオーナフィールドの配列
    - Map ファームオーナフィールド
        - String 名前(name)
        - String 値(value)

返り値
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - String オーナ UUID(uuid)

失敗した場合は、復帰コード以外は返りません。

注意
自分のものしか作れません。

update_owner

オーナの情報を更新します。

説明

update_owner は、オーナの情報を更新します。

- String ユーザ UUID
- String パスワード
- Array オーナフィールドの配列
    - Map オーナフィールド
        - String 名前(name)
        - String 値(value)
          住所の場合の value は配列で「郵便番号」「都道府県」「市区町村」「住所１」「住所２」とそれぞれの「公開・非公開設定」を渡します。
          住所以外はそのまま値を value にセットします。
            - Array 住所の補助フィールドの配列
                - Map 補助フィールド
                    - String 補助フィールド名(variable_name)
                    - String 値(value)
                    - Boolean 公開・非公開設定(private)
        - Boolean 公開・非公開設定(private)
          住所以外はオーナフィールドとして設定します。

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -2: 認証エラー
    - Integer 更新に成功したフィールドの数(count)

count は成功した場合のみ返ります。

注意

自分のものしか更新できません。

get_owner_fields

オーナフィールドの一覧を得ます。

説明

get_owner_fields は、オーナフィールドの一覧を得ます。
引数はありません。

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
    - Array オーナフィールドの配列(fields)
    - Map フィールド
        - String フィールド名(name)
        - String フィールドの型(type)
          LIST, RADIO, CHECKBOX, STRING, INTEGER, DATE, LOCATION
        - Integer または String フィールドの値
        - Array 選択肢
          フィールドの型が選択肢を持つ場合に存在する
            - String 選択肢のラベル
            - String 選択肢の値

注意

失敗することはありません。

get_prefectures

入力可能な住所のリストを得ます。

説明

get_prefectures は、入力可能な住所のリストを得ます。
引数はありません。

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
    - Array 都道府県の配列(prefectures)
      - Map 都道府県
        - Integer 都道府県 ID(id)
        - String 都道府県名(name)
    - Map 市区町村の配列(cities)
      - Map 都道府県ごとの市区町村(各都道府県名)
        - Array 市区町村のリスト
            - Map 市区町村
                - String 市区町村名(name)

注意

失敗することはありません。

get_owner_id

オーナの id を得ます。

説明

- String ユーザ UUID
- String パスワード

返り値

- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
      -2: 認証エラー
    - Integer Id(id)

失敗した場合は、復帰コード以外は返りません。

注意

自分の情報しか得られません。

自動承認 API

概要

自動承認 API は、利用者が所有するオーダーに対する自動応答の条件を参照、設定するものです。

API

自動承認に関する API には、以下のものがあります。

- get_auto_accept_order_setting
  現在の自動承認設定状態を取得
- update_auto_accept_order_setting
  自動承認設定状態を更新

get_auto_accept_order_setting

概要

現在の自動承認設定状態を取得します。

説明

get_auto_accept_order_setting は、ユーザ UUID とパスワードを与えてそのユーザの自動承認状態を返します。

- String ユーザ UUID
- String パスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 自動承認設定情報がない
    -2: 認証エラー
  - Boolean 商用利用するオーダー(commercial_use)
  - Boolean 商用利用しないオーダー(non_commercial_use)
  - Boolean 第三者提供するオーダー(third_party_use)
  - Boolean 提供者が未定でも承認する(undefined_third_party)
  - Boolean 第三者提供しないオーダー(non_third_party_use)
  - Boolean ポイント付与ありのオーダー(reward_points)
  - Boolean ポイント付与なしのオーダー(non_reward_points)

update_auto_accept_order_setting

概要
自動承認設定状態を更新します。

説明
update_auto_accept_order_setting は、ユーザ UUID とパスワード、自動承認の設定状態を与えてそのユーザの自動承認状態を更新します。

- String ユーザ UUID
- String パスワード
    - Boolean 商用利用するオーダー
    - Boolean 商用利用しないオーダー
    - Boolean 第三者提供するオーダー
    - Boolean 提供者が未定でも承認する
    - Boolean 第三者提供しないオーダー
    - Boolean ポイント付与ありのオーダー
    - Boolean ポイント付与なしのオーダー

返り値
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 自動承認設定情報がない
      -2: 認証エラー

通知 API

概要

通知 API は、通知を受け付けることができるデバイスのための API です。

API

通知についての API には、以下のものがあります。

- get_notification_setting
  通知設定の取得
- upadate_notification_setting
  通知設定の更新

get_notification_setting

概要

通知設定を取得ます。

説明

get_notification_setting は、ユーザ UUID とデバイス UUID 与えてそのデバイスの通知設定を取得します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - Map 通知設定(notifications)
    - Boolean 新着のオーダーが来た時(new_order)
    - Boolean 自動承認が行われた時(auto_accept_order)
    - Boolean データ取得の開始時(start_collect_data)
    - Boolean 審査結果に変更があった時(receive_review_result)
    - Boolean データ取得の終了時(end_collect_data)
    - Boolean ポイントが確定した時(fixed_point)

upadate_notification_setting

概要
通知設定の更新を行います。

説明
upadate_notification_setting は、ユーザ UUID とデバイス UUID と通知設定与えてそのデバイスの通知設定を取得します。

- String ユーザ UUID
- String パスワード
- String デバイス UUID
- Map 通知設定
    - Boolean 新着のオーダーが来た時(new_order)
    - Boolean 自動承認が行われた時(auto_accept_order)
    - Boolean データ取得の開始時(start_collect_data)
    - Boolean 審査結果に変更があった時(receive_review_result)
    - Boolean データ取得の終了時(end_collect_data)
    - Boolean ポイントが確定した時(fixed_point)

返り値
- Map 返り値
    - Integer 復帰コード(code)
      0: 成功
      -1: 失敗
    - String デバイス UUID(uuid)

データ出力 API

概要

レストランに配送されたデータを外部に取り出すためのものです。

API

データサブシステムからデータを取り出すための API は、以下のものがあります。

- get_output_data
  データを取り出します
- get_recipe_farms
  データの取得元ファームに関する情報を得ます
- clear_output_data
  出力バッファのデータを削除します

get_output_data

概要

出力バッファにあるデータを取り出します。

説明

get_putput_data は、出力バッファにあるデータを取り出します。

名前	型	意味
user_uuid	String	ユーザ UUID
login_name	String	ログイン名
password	String	パスワード
session_key	String	セションキー
recipe_uuid	String	レシピ UUID
keep	String	データを保持するかどうか（'true' or 'false'） 'false'　残さない 'true'　残す
limit	Integer	最大取り出し件数 1 度のアクセスで取り出す最大件数を指定します。デフォルトは 1000 です。

format	String	出力フォーマット データ出力の形式を指定します。デフォルトは'JSON'です。XML も指定可能です。
from	String	出力開始時刻 デフォルトは出力バッファの先頭からです。
to	String	出力終了時刻 デフォルトは出力バッファの末尾までです。

このファンクションでは、以下の 3 とおりの認証方法が提供されています。

- ユーザ UUID とパスワードによる方法
- ログイン名とパスワードによる方法
- セションキーによる方法

呼び出し時に必要なものを指定することにより、認証方法が選択されます。

返り値

成功した場合は、データが指定したフォーマットで返ります。失敗した場合は、標準の返り値で通知されます。

名前	型	意味
code	Integer	復帰コード
result	Text	出力データ

get_recipe_farms

概要

データの取得元ファームに関する情報を得ます。

説明

Map オプション

- String ユーザ UUID(user_uuid)
- String ログイン名(login_name)
- String パスワード(password)
- String セションキー(session_key)

- String レシピ UUID(recipe_uuid)
- String 出力フォーマット(format)
  データ出力の形式を指定します。デフォルトは'JSON'です。
  'XML' XML
  'JSON' JSON

認証方法には、

- ユーザ UUID とパスワードによる方法
- ログイン名とパスワードによる方法
- セションキーによる方法

があります。指定されたものに従って選択されます。

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
    -2: 認証エラー
  - String エラー説明(reason)

clear_output_data

概要

出力ファッバにあるデータを削除します。

説明

clear_output_data は、出力バッファにあるデータを削除します。
get_output_data で keep を指定した場合の後処理に使うことを想定しています。

呼称	型	意味
ユーザ UUID	String	
パスワード	String	
レシピ UUID	String	ダウンロードするデータのレシピを指定する
オプション	Map	ダウンロードのオプション。詳細後述

オプション

名前（タグ）	意味	型
from	出力開始時刻 デフォルトは出力バッファの先頭からです。	String
to	出力終了時刻 デフォルトは出力バッファの末尾までです。	String

返り値

Map 返り値

- Integer 復帰コード(code)
  0: 成功
  -1: 失敗
  -2: 認証エラー
- String エラー説明(reason)

レシピ API

概要

レシピ API は、所有するレシピ情報の参照ができます。

API

レシピ情報についての API には、以下のものがあります。

- get_recipe_list
  所有するレシピの一覧を取得します。
- get_recipe
  レシピ情報を取得します。

get_recipe_list

概要

所有するレシピの一覧を取得します。

説明

get_recipe_list は、ユーザ情報を元にそのユーザが所有するレシピの UUID の一覧を返します。

- String ユーザ UUID
- String パスワード

返り値

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - Array レシピ UUID の配列(uuids)
    - String レシピの UUID

get_recipe

概要

レシピの内容を得ます。

説明

get_recipe はレシピ UUID を指定して、レシピの内容を得ます。

- String ユーザ UUID
- String パスワード
- String レシピ UUID
  レシピに付与された UUID

返り値

成功すると、レシピの内容が Map で返ります。
レシピの内容は以下のようになります。

- Map 返り値
  - Integer 復帰コード(code)
    0: 成功
    -1: 失敗
  - String レシピ UUID(uuid)
  - String レシピ名(name)
  - String 説明(description)
  - String 利用目的(purpose)
  - String データ収集開始日時(collection_period_start)
  - String データ収集終了日時(collection_period_end)
  - String 募集開始日時(application_period_start)
  - String 募集終了日時(application_period_end)
  - Boolean 第三者提供(third_party)
  - String 第三者提供：提供先(third_party_name)
  - Boolean 商用利用(commercial_use)
  - String 審査(judge_type)
    none: 審査なし
    judge: 審査あり
  - String 募集数(quota_of_farms)
  - Boolean ポイント提供あり(point_supply_enabled)
  - String ポイント率(point_rate)
  - Integer 最低ポスト回数(minimum_post_count_to_point)
  - Integer ボーナスポイント(bonus_point)
  - Integer 完了ポスト回数(minimum_post_count_to_complete)

- String 最大獲得ポイント(maxmum_gain_point)
- String 最大ポスト数(maximum_posts)
- Array 収集情報(collection_sensor_data)
  - String 仮想センサー名(virtual_sensor_name)
  - String デバイスクラス(device_class)
  - String デバイスセンサー(device_sensor)
  - String 出力単位(meta_level)
  - String 設置場所種別(location_type)
    undefined 未設定
    fix 固定
    movable 移動
  - String 設置位置：都道府県(location_point_prefectures)
  - String 設置位置：市区町村(location_point_city)
  - String 設置位置：アドレス(location_point_address)
  - String 設置場所(location_in_out)
    undefined 屋内/屋外
    indoor 屋内
    outdoor 屋外
- String オーダー発行単位(order_issue_unit)
- String データ取得方法(event_trigger)
- Integer 計測間隔:秒(time_interval)
- String 計測間隔単位(timer_unit)
- Integer 計測データ最大受信遅延時間(time_maximum_latency)
- String 計測データ最大受信遅延時間単位(time_maximum_latency_unit)
- Integer 計測時刻の許容範囲(time_permissible_range)
- String 計測時刻の許容範囲単位(time_permissible_range_unit)
- Boolean 自分にだけオーダーを発行する(private_recipe)
- Array 対象となる提供者の条件(farm_owner_conditions)
  - String 項目名(field)
  - String 条件名称(function_name)
  - String 条件種別(function_type)
  - Map 値(value) *1
- String 取消日時(canceled_at)
- String 更新日時(updated_at)
- String 削除日時(deleted_at)

*1 条件種別により返される Map の内容が異なります。

条件種別

StringContainAnd

- Map 値(value)
    - Array 文字列の配列(include)
        - String 文字列

LocationMatch

- Map 値(value)
    - String 都道府県(prefecture)
    - String 市区町村(city)
    - String アドレス(address)

CheckboxIn/RadioIn

- Map 値(value)
    - Array 選択された値(selected)
        - String 文字列

DateRange

- Map 値(value)
    - String 開始日(begin)
    - String 終了日(end)

JSON over HTTP ゲートウェイとのやりとり

現在のデータサブシステムは、HTTP で JSON をやりとりするゲートウェイが用意されています。基本的には既に述べた API を JSON と HTTP を使うというように読み換えることで使えますが、よく使われる API だと思われるので、あらためて解説します。

パラメータの意味等の詳細については、API 説明書（内部 API）を参照して下さい。

アクセスする URL

EverySense の JSON over HTTP ゲートウェイは、以下の URL です。

```
http://api.every-sense.com:7001/
```

また、HTTPS の場合は、

```
https://api.every-sense.com:8001/
```

です。

用意されている API

以下の API が用意されています。

機能名	メソッド	意味
device_data	POST	デバイスからデータを送ります。内部的には put_message です。
device_data	GET	デバイスから送られたデータを読み出します。この API は主にデバイスデータを確認するために使われます。

recipe_data	GET	レシピに届いたデータを読み出します。内部的には、get_output_data です。
session	POST	認証を行い、セションを作成し、セションキーを取得します。
session	DELETE	セションを無効にします。

これら以外の API については、API 説明書にある API と同じ名前で POST することで利用可能ですが、現在整理中です。

API と引数

POST device_data

呼び出し方

```
/device_data/<デバイス UUID>
```

データ

センサーデータを JSON で表現したものをデータとして渡します。

返り値

正しく処理された場合は、put_message の返り値、すなわち成功した場合の標準の返り値が JSON 形式で返ります。具体的には、以下のような JSON です。

```
{
     code: “0”
}
```

ゲートウェイ内でエラーが発生した場合は、’code’ が -10、 ‘message’ が「エラー理由」の JSON が返ります。それ以外のエラーも ‘code’ が負数となって、エラーメッセージ等が返ります。一般的には、以下のような形式の JSON です。

```
{
      code: "<復帰コード>",
      reason: "<エラー発生の理由>",
      message: "<エラーメッセージ>",
      trace: "<サーバプロセスのバックトレース>"
}
```

GET device_data

呼び出し方

```
/device_data/<デバイス UUID>?<オプション>
```

オプション

以下のものが指定可能です。

オプション	意味	説明
user_uuid	ユーザ UUID	認証情報といてユーザを識別するのに使います。
login_name	ログイン名	認証情報としてユーザを識別するのに使います。 user_uuid と同時に指定した場合、user_uuid を優先します。
password	パスワード	認証のためのパスワードです。user_uuid または login_name を指定した場合に使います。 平文の上、URL で見えてしまうので、注意して使って下さい。
session_key	セションキー	POST session で作成したセションキーを指定します。 キーの無効化も簡単なので、パスワードを使った認証よりも安全です。
limit	取得上限件数	一度に取得するデータ件数の上限を指定します。 デフォルトは 1000 なので、それ以上取得したい場合は、明示します。
from	取得開始時刻	取得するデータの開始時刻です。
to	取得終了時刻	取得するデータの終了時刻です。

返り値

正しく処理され場合は、put_message の返り値、すなわち成功した場合の標準の返り値が JSON 形式で返ります。具体的には、以下のような JSON です。

```
{
      code: "0"
}
```

ゲートウェイ内でエラーが発生した場合は、’code’ が -10、‘message’ が「エラー理由」の JSON が返ります。それ以外のエラーも ‘code’ が負数となって、エラーメッセージ等が返ります。一般的には、以下のような形式の JSON です。

```
{
      code: "<復帰コード>",
      reason: "<エラー発生の理由>",
      message: "<エラーメッセージ>",
}
```

ただし、エラーの種類によっては、全てがない場合もあります。

GET recipe_data

呼び出し方

```
/recipe_data/<レシピ UUID>.<フォーマット>?<オプション>
```

オプション

以下のものが指定可能です。

オプション	意味	説明
user_uuid	ユーザ UUID	認証情報としてユーザを識別するのに使います。
login_name	ログイン名	認証情報としてユーザを識別するのに使います。 user_uuid と同時に指定した場合、user_uuid を優先します。

password	パスワード	認証のためのパスワードです。user_uuid または login_name を指定した場合に使います。 平文の上、URL で見えてしまうので、注意して使ってください。
session_key	セションキー	POST session で作成したセションキーを指定します。 キーの無効化も簡単なので、パスワードを使った認証よりも安全です。
limit	取得上限件数	一度に取得するデータ件数の上限を指定します。 デフォルトは 1000 なので、それ以上取得したい場合は、明示します。
from	取得開始時刻	取得するデータの開始時刻です。
to	取得終了時刻	取得するデータの終了時刻です。
keep	データの保管	‘true’を指定すると、データ取得後にデータ削除を行いません。
inline	インライン指定	‘true’を指定すると、ファーム情報をインラインでデータに埋め込みます。 当然、公開されているもののみです。
format	フォーマット	正常に処理された時に返すデータのフォーマットを指定します。通常、フォーマットは URL のパス部で指定されますが、オプションで変更することも出来ます。 現在指定可能なものは、XML, JSON です。

返り値

正しく処理された場合は、HTTP ステータスが 200 で、指定したフォーマットでレシピに収集されたデータが出力されます。

何らかのエラーが発生した場合、HTTP ステータスが 500 で、JSON で以下の形式のメッセージが出力されます。

```
{
      code: “-10”,
      message: “<エラーメッセージ>”,
}
```

POST session

呼び出し方

```
/session
```

データ

以下のデータを JSON 形式で渡します。

オプション	意味	説明
user_uuid	ユーザ UUID	認証情報としてユーザを識別するのに使います。
login_name	ログイン名	認証情報としてユーザを識別するのに使います。 user_uuid と同時に指定した場合、user_uuid を優先します。
password	パスワード	認証のためのパスワードです。user_uuid または login_name を指定した場合に使います。 平文の上、URL で見えてしまうので、注意して使って下さい。

返り値

正しく認証された場合は以下のような JSON が返ると共に、セションが生成されます。このセションキーを使って、認証が必要な API をアクセスします。

```
{
        code: “0”,
        session_key: “<セションキー>”,
}
```

エラーのあった場合は、以下のような JSON が返ります。

```
{
        code: “<復帰コード>”,
        message: “<メッセージ>”,
}
```

DELETE session

呼び出し方

```
/session/<セションキー>
```

返り値

与えたセションキーに対応するセションが存在した場合は、

```
{
        code: 0
}
```

が返り、セションキーは無効になります。
与えたセションキーに対応するセションが存在しない場合は、

```
{
        code:  “-1”
        reason:  “session not found”
}
```

が返ります。

センサーデータ

センサーデータの種類

基本的なセンサーについて、そのデータ形式を規定します。

- 気温センサーデータ
- 水温センサーデータ
- 湿度センサーデータ
- 土壌温度センサーデータ
- 電気伝導度センサーデータ
- 降水量センサーデータ
- 気圧センサーデータ
- 風向センサーデータ
- 風速センサーデータ
- 近接センサーデータ
- 加速度センサーデータ
- ジャイロセンサーデータ
- 方位センサーデータ
- 磁気センサーデータ
- 騒音センサーデータ
- 歩数センサーデータ
- モーションアクティビティセンサーデータ
- 位置センサー

センサーデータを表現する型について

特定のプログラム言語等に依存しないでデータ形式を表現するために、以下の抽象型を定義します。

- Number
  数値（スカラー）を表現します
- GeoLocation
  地理的位置を表現します
- Dimension
  寸法や距離といった、物理的な大きさを表現します

- TimeStamp
  日付を含む時刻を表現します
- Text
  文書を表現する文字列で、改行等を含みます
- String
  文字列で、改行等を含みません
- Symbol
  名前です
- Enumerate
  １つ以上の Symbol からなる集合の Symbol を値とします

共通フィールドについて

以下のフィールドは、センサー共通のフィールドです。ただし、センサーの性質によっては存在しないこともあります。

名前	意味	型
location	センサー設置場所	Location
unit	単位	String
accuracy	精度	Number
memo	覚書	Text
at	値取得時刻	TimeStamp

精度は相対精度として、標準単位は”percent”です。
時刻はタイムゾーン情報を含めます。ない場合は UTC とみなします。

気温センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”degree_Celsius”です。

水温センサーデータ

名前	意味	型
depth	センサー設置水深	Dimension
value	センサー値	Number
depth_unit	センサー設置水深の単位	String

標準の単位は”degree_Celsius”、水深の単位は”m”です。

湿度センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”%RH”です。

土壌温度（soil temperature）センサーデータ

名前	意味	型
depth	センサー設置深さ	Dimension

value	センサー値	Number
depth_unit	センサー設置深さの単位	String

標準の単位は”degree_Celsius”です。センサー設置深さの単位は”cm”です。

電気伝導度 EC センサーデータ

名前	意味	型
depth	センサー設置深さ	Dimension
value	センサー値	Number
depth_unit	センサー設置深さの単位	String

標準の単位は”mS/mm”、設置深さの単位は”cm”です。

降水量

名前	意味	型
value	センサー値	Number

標準の単位は”mm”です。

気圧センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”hPa”です。

風向センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”degree”です。

風速センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”m/s”です。

近接センサーデータ

名前	意味	型
value	センサー値	String(“on” or “off”)

単位を持ちません。

加速度センサーデータ

名前	意味	型
values	[センサー値 X,センサー値 Y,センサー値 Z]	Number Array

標準の単位は”m/s^2”です。
iPhone 等では”G”で取得されると思いますが、変換するか unit に”G”を指定します。

ジャイロセンサーデータ

名前	意味	型
values	[センサー値 roll,センサー値 pitch,センサー値 yaw]	Number Array

標準の単位は”deg/s”です。

方位センサーデータ

名前	意味	型
value	方位	Number
unit	方位単位	String
north	北の種類	String (“magnetic” or “true)

標準の単位は”degree”です。北の種類の標準は”true”です。

磁気センサーデータ

名前	意味	型
values	[センサー値 X,センサー値 Y,センサー値 Z]	Number Array

標準の単位は”nT”です。

騒音センサーデータ

名前	意味	型
value	センサー値	Number

標準の単位は”dB”です。

歩数センサーデータ

名前	意味	型
value	センサー値	Number

単位は”step”のみです。

モーションアクティビティセンサーデータ

名前	意味	型
value	センサー値	String

単位はありません。

位置センサー

名前	意味	型
values	[経度,緯度,標高]	Dimension
datum	測地系	String

標準の単位は”degree”，測地系は”WGS84”です。

出力されるデータの形式

既に説明したように、データサブシステムから出力可能なデータは、

- レシピに振り分けされたセンサーデータ
- デバイスから送信されたセンサーデータ

となります。基本的に、センサーデータの内容は同じですが、前者は振り分けの際に、

- ファームで公開されているデータ
- デバイスを識別するための UUID
- センサーのクラス名（センサーの種類の名前）
- センサーデータのクラス名（データの単位の名前）

等が付加されています。

実際に出力されるデータは、このデータをそれぞれのデータフォーマット（JSON, XML 等）に合わせて serialize されたものです。

以下にその詳細を示します。

振り分けされたデータの形式（get_output_data で取得されるデータ）

振り分けされたデータは、以下に示す形式のエントリの配列を 1 つのデバイスのデータレコードとします。

名前	意味	説明	型
device_uuid	デバイスを表現する UUID	この UUID が同じデータは、同じデバイスから出力されていることが保証されています。レシピが異なる場合は、同じデバイスでも別の UUID となります。	String(36)
sensor_name	センサー名	レシピで設定した仮想的なセンサー名。	String

sensor_class_name	センサークラス名	センサーを表現するクラス名、詳細はセンサーリポジトリを参照のこと。	String
data_class_name	センサーデータクラス名	測定値を表現するクラス名、たとえば”Accelation（加速度）”といったもの。詳細はセンサーデータクラス名の解説を参照のこと。	String
data	センサーデータ	センサーのデータ。形式についてはセンサーデータの解説を参照のこと。	Map

sensor_name は、レシピ作成の時に指定します。

レシピに従って収集されるデータは、「sensor_name を持った仮想的なセンサーを持ったデバイスからのデータ」という考え方をします。具体的にファームのどのセンサーがマッピングされるかについては、レシピの要求とファームの内容とのマッチングによって決定されます。

デバイスからの送信データ（get_message で取得されるデータ）

デバイスから送信されたデータは、以下に示す形式のエントリの配列を 1 回のセンスのデータレコードとします。

名前	意味	説明	型
sensor_name	センサー名	デバイスローカルなセンサー名。	String
data	センサーデータ	センサーのデータ。形式についてはセンサーデータの解説を参照のこと。	Map

センサーのリポジトリ

センサーの名寄せを行うためのリポジトリについて記述したもので、以下のことについて記述しています。

- 基本的な考え方
- 分類
- 対象

基本的な考え方

- センサークラスの呼び方は、[<parameter>] <measurement> of <target> [<modifier>]　の形式である
- センサークラスの名前は、「呼び方」を元に構成される
- リポジトリにない measurement, parameter, target, modifier は、随時追加される
- センサーの値は１つ以上のフィールドを持ち、同じセンサークラスのセンサーは同じフィールドを持つ

センサークラスの例

名前	意味	型
temperature of water	Water Temperature	水温
PM2.5 number of air	AirPM25Number	空気中の PM2.5 の数
location of person	PersonLocation	人の位置

measurement の語彙

名前	シンボル
温度	temperature
湿度	humidity

圧力	pressure
変位	displacement
角度	angular
位置	location
照度	illuminance
速度	velocity
加速度	acceleration
頻度	frequency
放射線	radiation
濃度	concentration
数	number

terget の語彙

名前	シンボル
空気	air
水	water
方向	direction
土壌	soil
歩	step
人	person