

富山市 センサーネットワーク 利用ガイド

2020年4月

目次

1. はじめに
 1. はじめに
 2. 用語の説明
 3. システム構成
2. センサーネットワークについて
 1. LoRaWANシステム概要
 2. データ利活用基盤概要
 3. ダッシュボード概要
3. 利用を開始するために
 1. 利用開始フロー
 2. 作成資料一覧
 3. ネットワークサーバへの登録
 4. データモデル設計
 5. デバイス付与情報登録
4. アプリケーション開発
 1. 認証について
 2. 各APIの利用方法
 3. 地図データの利用方法

1. はじめに

1-1. はじめに

■ 本資料について

富山市センサーネットワークはFIWARE（データ利活用基盤）とLoRaWANのアダプタを持つシステム構成です。
本資料は富山市センサーネットワーク全体の概要と、利用を始めるにあたって必要な作業について記載しています。

■ 説明内容

- データ利活用基盤の概要、機能について説明
- 富山市センサーネットワークを利用するための作業について説明
- アプリケーションにて使用する基本的なAPIの使い方を説明

■ 前提知識

- WEBアプリケーションの開発経験が必要
※専用アプリケーションを開発する場合

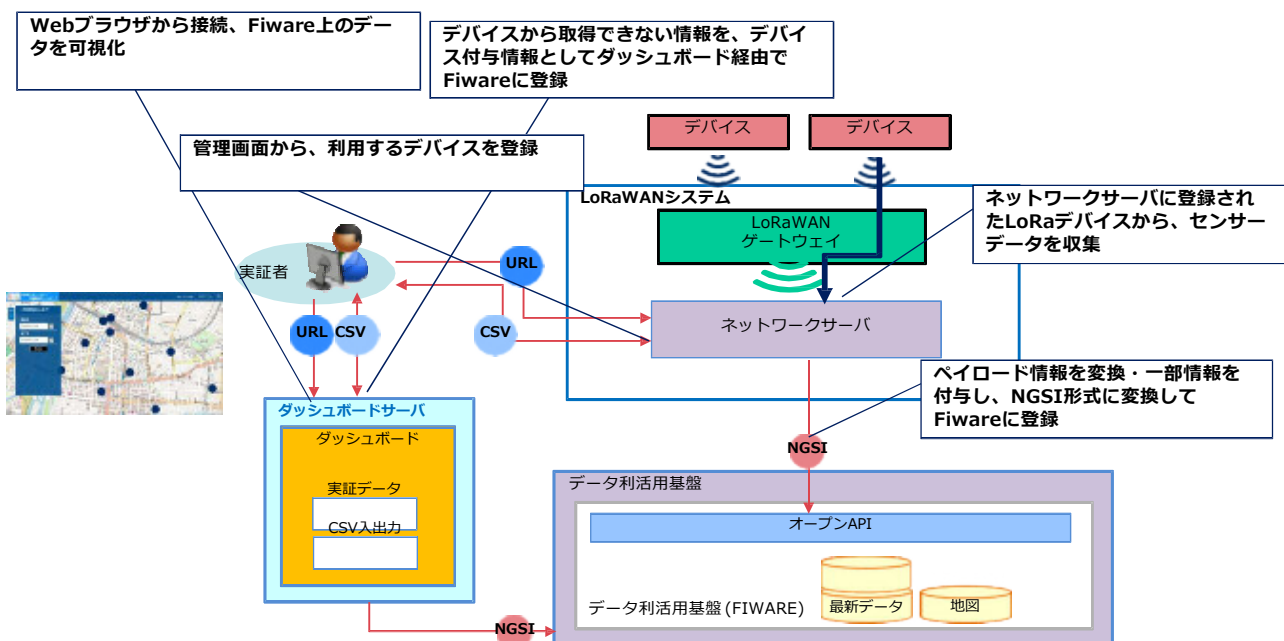
1-2. 用語の説明

用語・略語	説明
FIWARE	公共サービスを提供する自治体や企業などの業種を越えたデータの利活用やサービス連携を促すためのオープンソース・ソフトウェア（OSS）として開発され、標準化されたオープンAPIを持つIoTサービスプラットフォーム
LoRaWAN	LoRaは、LoRa Allianceによって定められた、LPWA(Low Power Wide Area Network)と呼ばれる低消費電力で広域のエリアをカバー（都市部で2~10km）する無線通信規格の一つ。
NGSI	Next Generation Service Interfaces。NECが2009年にOMAに提案し、標準化された次世代サービスインタフェース
コンテキスト	IoTプラットフォームのシステムで受け渡しされるデータ
API	Application Programming Interface。ソフトウェアコンポーネントが互いにやりとりするのに使用するインタフェースの仕様
JSON	JavaScript Object Notation。軽量のデータフォーマット、構文はJavaScriptにおけるオブジェクトの表記法をベースとしている
OMA	Open Mobile Alliance。モバイルによるWebアプリケーションの技術標準化を目指す業界団体
OSS	オープンソースソフトウェア。利用者の目的を問わずソースコードを使用、調査、再利用、修正、拡張、再配布が可能なソフトウェアの総称
RESTful API(REST API)	Representational State Transfer。Webシステムのアーキテクチャスタイル
GeoJSON	JSONを用いて空間データをエンコードし非空間属性を関連付けるファイルフォーマット

5

1-3. システム構成

富山市センサーネットワークは以下のシステム構成となっております。
※デバイスは実証事業者様にて準備頂く必要がございます。



6

2. センサーネットワークについて

2-1. LoRaWANシステム概要

- 居住区の98%をカバーするLoRaWANネットワークを構築
- 免許不要で長距離、低消費電力の通信が可能

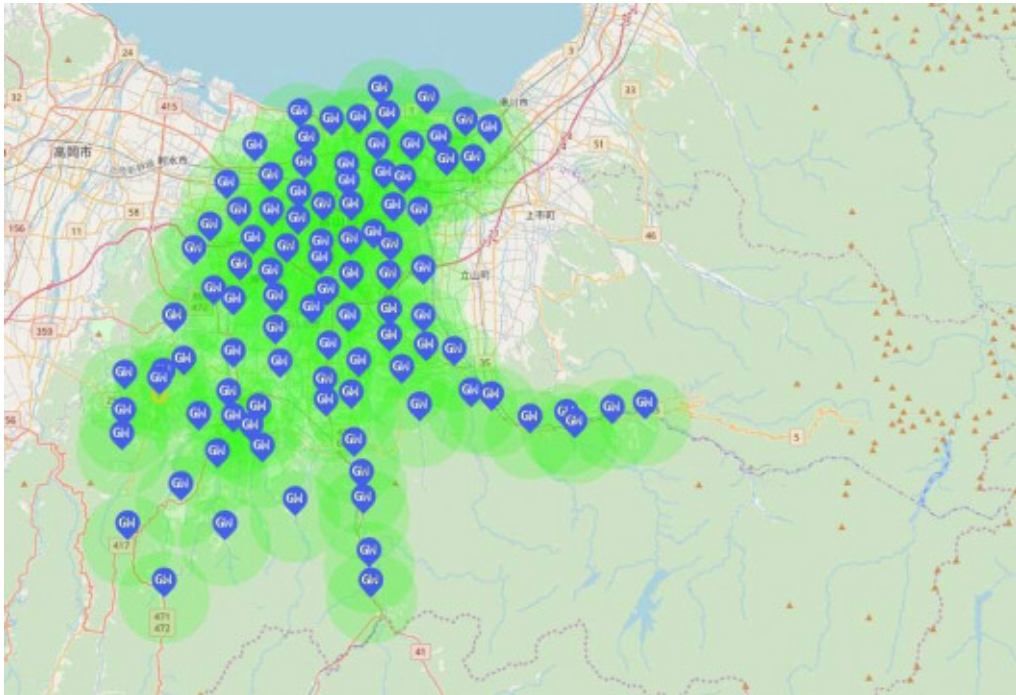
1. 対応LoRaデバイスの規格 LoRaWAN (Spec1.0.2)に準拠しているもの
富山市環境で動作確認済みデバイス：グリーンハウス社製 GH-GPSTLRA (ClassA、Spec1.0.2)
2. 性能(1GWあたりの収容数)
最大 約700台/分 (※)
(※)1分周期で送信するデバイスを1,200台接続させた際の実績データに基づく
3. 制約条件
 - デバイスは、ClassA、B、Cいずれも利用可能
 - デバイスがMACコマンドに対応していない場合は、LoRaネットワークサーバ提供機能 (ADR、チャンネル変更) が利用できません。
 - デバイスのデータ送信周期や接続台数はLoRa網全体の負荷設計に関わるため利用前に必ず申請する (予測も含め) ようにしてください
 - デバイスのアドレス (Devaddr)は、OTAA/ABPともに利用可能ですが、ABPを利用する場合、指定するDevaddrの範囲であらかじめ設定をお願いします。
 - 利用可能な周波数帯 (チャンネル)は、指定した範囲でのご利用となります。

2-1. LoRaWANシステム概要

4. 利用可能エリア

富山市内のLoRaWANシステムは以下の範囲をカバーしています。

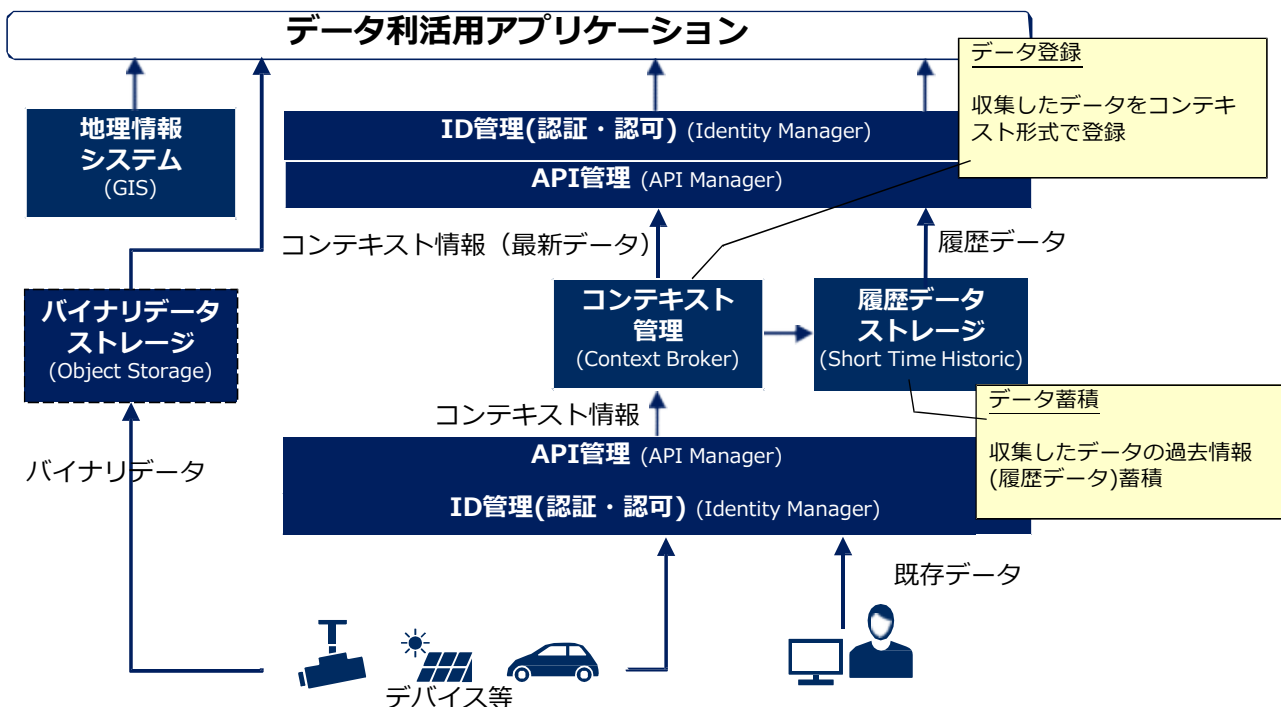
※周辺の建物等の条件により電波が届かない箇所がある可能性があります。



9

2-2. データ利活用基盤概要

- データ利活用型のプラットフォームであるFIWAREをベースに環境を構築
- FIWARE の標準機能(コンポーネント) + NEC拡張機能 によって実現



10

2-2. データ利活用基盤概要

データ利活用基盤の機能を以下に記載します。

機能名	概要
コンテキスト管理	OMA NGSIインターフェースを介したコンテキスト情報（最新データ）の登録・蓄積・参照、Pub/Sub機能を提供
履歴データストレージ	コンテキスト管理情報の変化を履歴データとして蓄積・参照する機能
リアルタイム分析	収集データをリアルタイム分析し、結果を出力する機能
地図基盤 (GIS)	アプリケーションへ地理情報を提供する機能
バイナリデータストレージ	画像・動画などのバイナリデータを管理するストレージ機能
ID管理 (認証・許可)	管理機能やAPIへのアクセス権限をユーザIDで制御する認証・認可機能
API管理	Web APIの管理機能

11

2-3. ダッシュボード概要

- データ利活用基盤に蓄積された情報をアプリ内にて表示
- バイナリデータ、履歴データ、地図データを利用した情報の可視化

最新情報表示

各コンテキストの最新情報（現在の状態）を表示する。

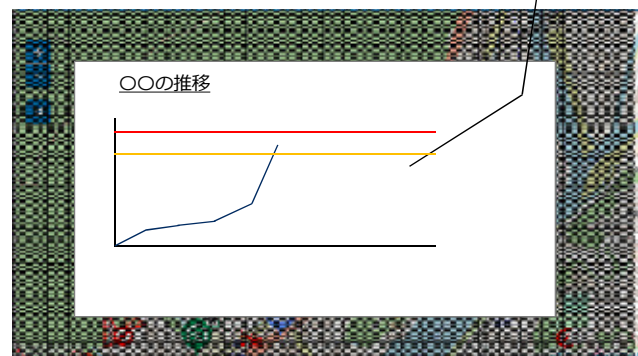
CSVエクスポート

データ利活用基盤内の情報をCSV形式で出力する。



履歴データ表示

収集したコンテキスト情報の推移をグラフで表示する。



12

2-3. ダッシュボード概要

ダッシュボードの主な機能を以下に記載します。

機能名	概要
地図表示	地図を表示する。※[+][-]ボタン、マウスホイールによる縮尺変更が可能
最新データ表示	データ活用基盤内の情報の最新データを地図上にアイコンをプロットする ※アイコンの種類ごとに、地図上にプロットするアイコンのON/OFFを切り替え可能
最新データ詳細情報表示	最新データの1データ（1エンティティID）の詳細情報（属性情報）を表示する
最新データ絞り込み	詳細情報（属性情報）で条件指定し、地図上にプロットするデータ（エンティティID）を絞り込む
最新データCSV出力	データ1種類の最新データの全データをCSVに出力する
履歴データCSV出力	1データ（1エンティティID）の内の1属性の履歴データをCSVに出力する
履歴データのグラフ表示	1データ（1エンティティID）の内の1属性の履歴データの推移をグラフに表示する

13

2-3. ダッシュボード概要

使用可能なブラウザは以下のとおりです。

No.	項目名	使用者	PC(Windows)			タブレット (iPad)
			Chrome	Internet Explorer	Microsoft Edge	Safari
1	ダッシュボード	利用者	○	○	○	○
2	パスワード変更画面	利用者	○	×	×	×

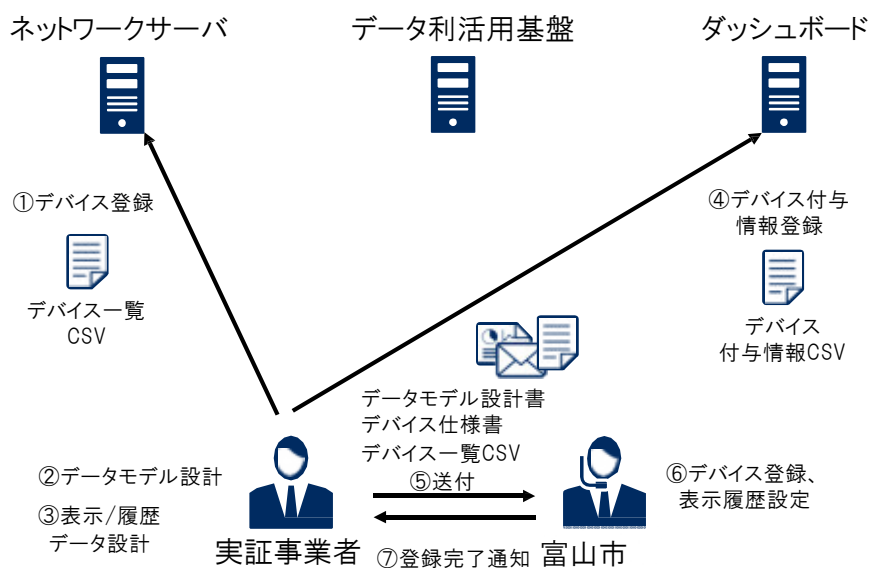
※2019年2月時点での最新バージョンで動作を確認します。

14

3. 利用を開始するために

3-1. 利用フロー※利用申請完了後

富山市センサーネットワークをご利用いただくため、以下の作業を実施いただく必要があります



3-2.作成資料一覧

前述のフローの中で、以下の資料を実証事業者様にて作成・提出いただく必要があります。

項番	作業名	作成物	説明
①	デバイス登録	デバイス一覧CSV	ネットワークサーバにデバイスを登録するためのCSVファイルとなります。ネットワークサーバ管理画面から、デバイスを登録いただく際に必要となります。
②	データモデル設計	データモデル設計書	データ利活用基盤、ダッシュボードの設定を実施し、デバイスからの情報をデータ利活用基盤に登録できるよう設定するための設計書となります。 データ利活用基盤で取り扱うデータの内容を設計いただきます。合わせて、ダッシュボード側での表示有無、CSV出力の有無等、富山県にてダッシュボードの設定をする際に必要な情報を設計いただきます。
③	表示/履歴データ設計		
④	デバイス付与信息登録	デバイス付与信息 CSV	デバイスから取得できない情報を、デバイス付与信息としてデータ利活用基盤に設定するためのCSVファイルとなります。

17

3-3. ネットワークサーバでの作業について

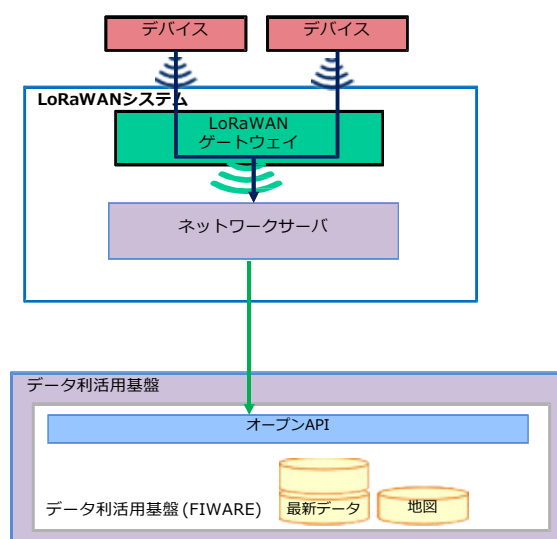
利用を開始する前に、ネットワークサーバに利用するデバイスを登録、設定を実施いただく必要があります。

1.デバイス登録

CSV、もしくはネットワークサーバのGUIにてデバイスを登録する必要があります。
デバイスのデータをネットワークサーバ上に蓄積するために必要な作業となります。

2.通知端末設定

CSV、もしくはネットワークサーバのGUIにて登録したデバイスの通知端末設定を実施いただく必要があります。
ネットワークサーバの情報をデータ利活用基盤にて収集・蓄積するために必要となります。



18

3-4. データモデル設計

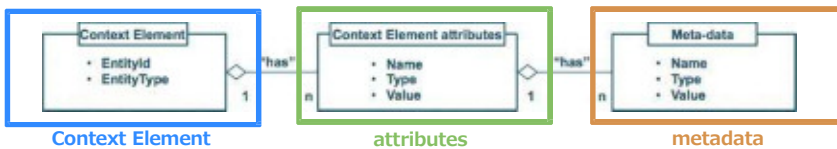
データ利活用基盤へのデータ登録・ダッシュボードの設定のため、実証事業者様にて「データモデル設計書」を作成いただきます。

データモデル設計書例（例：子供の位置情報）

エンティティ項目	値	説明
論理名	子供の位置情報	エンティティの論理名。NGSIには指定しない
type	ChildrenPosition	エンティティのタイプを指定する。固定値で左の値を指定する。
isPattern	false	エンティティ検索時に使用する。NGSIの定義では固定値、左の値を指定する。
id		URIと識別子の結合により一意のIDとする。 URI : 組織、グループなどをURIで表す。 下記を使用する。 jp.toyama.smartcity.position 識別子 : [Entity Type].[任意の識別子(連番、UUID等)]とする。 URIと識別子は、「.」により結合をする。[URI].[識別子] jp.toyama.smartcity.position.ChildrenPosition.1

項目名	attributes				meta			
	name	type	参考値	説明	name	type	参考値	説明
DeviceEUI	DeviceEui	xsd:string	00-00-00-FF-FE-00-00-00	DeviceEUIを格納する。				
GPS受信日時	ReceiveDate	xsd:dateTime	2018-12-12T12:12:00+09:00	GPS受信日時(GWがエンドデバイスからUplinkデータを受信した時刻)を格納する。 形式: iso8601 日本時間				
座標	Coordinate	geo:point	34.350806005890014, 134.0465497970581	位置情報を格納する。 形式: 緯度, 経度				
学校	SchoolName	xsd:string	〇〇小学校	学校の名称を格納する。				
学年	SchoolGrade	xsd:string	1年	学年を格納する				
更新日時	UpdatedDate	xsd:dateTime	2018-12-12T12:12:00+09:00	[GPS受信日時]と同値を格納する。				

NGSI データモデル



19

3-4. データモデル設計

ダッシュボードには作成したデータモデルのアイコンが表示されます

The screenshot shows a dashboard interface. On the left, there is a map with several colored icons representing different data models: purple for '子供の位置情報' (Child Location Information), blue for '学校教育施設' (School Education Facilities), and pink for '市役所・行政センター' (City Office/Administration Center). On the right, there is a menu with the following items: '子供の位置情報', '学校教育施設', and '市役所・行政センター'. A callout box points to the menu items, stating: 'メニューには、Context Element単位で「アイコン表示切替」項目を表示' (In the menu, the 'Toggle Icon Display' item is displayed at the Context Element level).

20

3-4. データモデル設計

詳細情報には、属性(attribute)ごとに属性名と値が表示されます。

The screenshot shows a mobile application interface. On the left is a map of a city area with several purple and blue markers. A callout box points to one of the markers with the text "attribute単位で詳細情報を表示". On the right is a data panel titled "子供の位置情報" (Child's Location Information). The panel contains the following fields:

- DeviceEUI: 00-00-00-FF-FE-00-00-00
- GPS受信日時: 2018-12-12T12:12:00+09:00
- 座標: 34.350806005890014, 134.0465497970581
- 学校: ○○小学校
- 学年: 1年
- 更新日時: 2018-12-12T12:12:00+09:00

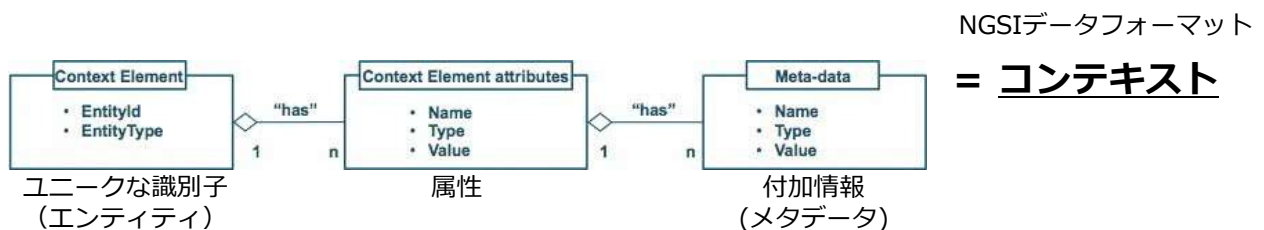
21

3-4. データモデル設計

データ利活用基盤ではNGSIというデータモデルを使用

データ利活用基盤で利用するデータモデルについて

- 利用者はNGSI(Next Generation Service Interfaces)というデータモデルにもとづいた形式で作成します。
- NGSIとは実世界上の物理オブジェクトを、エンティティ、属性、メタデータで表現するデータモデルであり、エンティティの定義を統一することでドメインを跨ぐサービスやアプリによるデータ相互運用が可能となります。



参考 : FIWARE NGSI APIv1 Walkthrough

https://fiware-orion.readthedocs.io/en/master/user/walkthrough_apiv1/

22

3-4. データモデル設計

例として「デバイス(センサー)」から取得したデータを取り扱う場合のデータモデル決定方法を記載する。

1. 取り扱うデータ項目を予め想定しておく。
「デバイス」データで必要な項目を抽出する。



2. サンプルデータを参考に構造（属性の構成、データの型）の定義を決定する

- データの種類名 : Device
- 個別のデバイスID : device-0001A
- Attribute : 取得するデータ名、データの型
※座標の型は、NGSIの制約に従いgeo:pointとして定義

3. データ項目に過不足がないか確認

データの種類やAttribute情報の参考として下記で定義した情報を利用することを推奨する。

FIWARE Datamodels : <https://www.fiware.org/developers/data-models/>

Schema.org : <http://schema.org/>

共通語彙基盤 : <https://goikiban.ipa.go.jp/>

23

3-5. デバイス付与情報登録

デバイスから取得できない情報を登録する場合、デバイス付与情報を登録する必要があります。

デバイス付与情報はダッシュボードからCSV入力機能で登録が可能です。

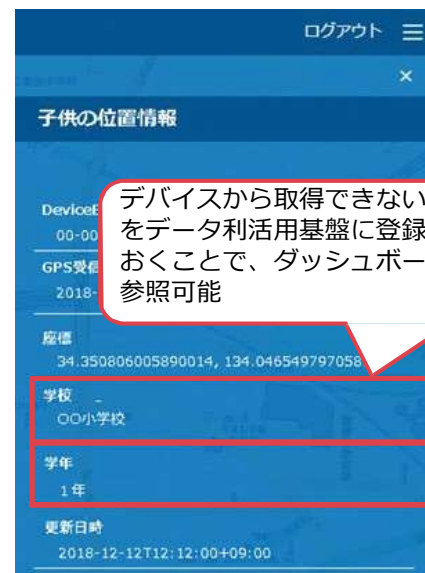
例

こどもにGPSデバイスを持たせて、位置情報を地図上に表示する場合。

位置情報はデバイスから取得できるが、こどもの小学校名や学年もデータ利活用基盤に登録したい。



小学校名・学年は「デバイス付与情報」として登録しておく必要がある。



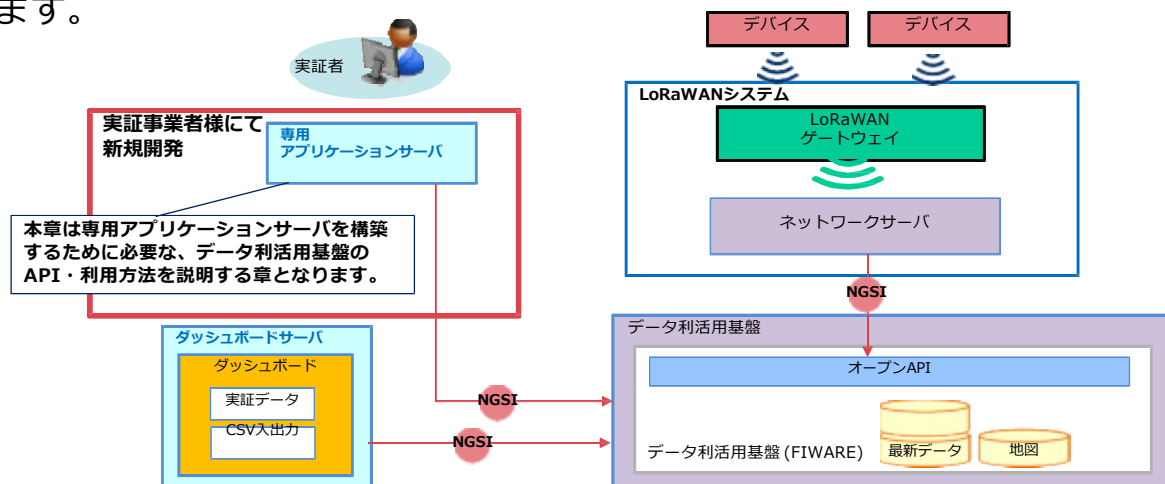
24

4. アプリケーション開発

4. アプリケーション開発

本章は、富山市センサーネットワークを、ダッシュボード以外のアプリケーションにて利用する開発者向けの内容となっております。

- データ利活用基盤上に収集したデータを参照する汎用的なアプリケーションとしてダッシュボードサーバが提供されています。より実証事業に特化したアプリケーションが必要な場合、実証事業者様にアプリケーションを開発し、データ利活用基盤を活用することが出来ます。



4-1. 認証機能

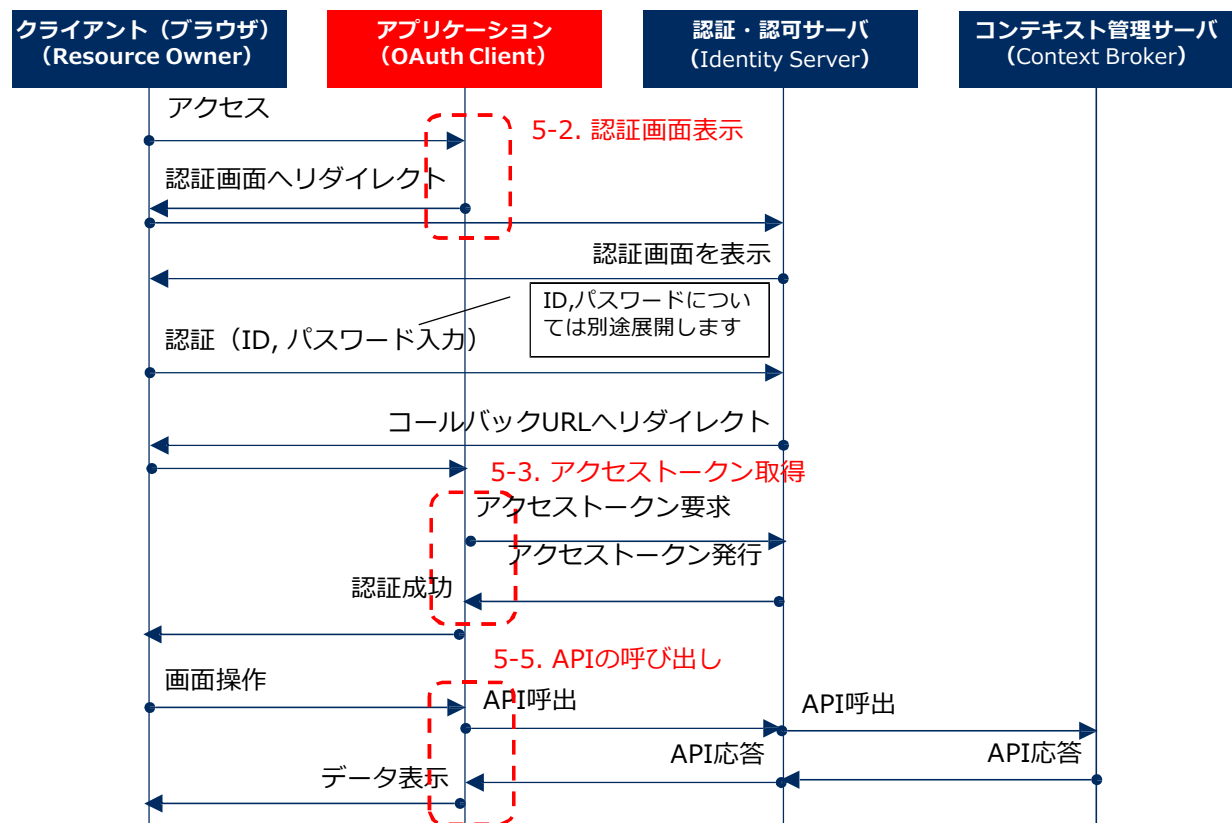
APIの利用を制御するためOAuth 2.0による認証機能が利用可能
 OAuth 2.0は、4通りの認証方法があり、利用者はアプリケーションが必要とするセキュリティレベルや実装方式等により認証方式を選択できます。

認証方式	説明
Authorization Code Grant	信頼関係にないWebアプリケーションに有効な認証方式。 アプリケーションには認証情報を渡さず、利用者自身が認可サーバの認証を受ける。Consumer Key, Consumer Secretを利用する。
Implicit Grant	JavaScriptのような、資格情報を秘密にできないプログラムに有効な認証方式。Consumer Secretを利用せず、Consumer Keyのみで認証する。
Resource Owner Credentials Grant	信頼関係(同一ドメイン内など)のあるWebアプリケーションに有効な認証方式。 アプリケーションに対し利用者が認証情報 (ID, パスワード) を提供する必要がある。
Client Credentials Grant	プログラム(バイナリ)に有効な認証方式。 利用者の認証情報 (ID, パスワード) は利用せず、アプリケーションの認証情報 (ConsumerKey, ConsumerSecret) を利用して認証する。

※各認証の詳細についてはアプリケーション開発ガイド（認証認可編）を参照。

27

4-1. 認証機能 認証シーケンス (Authorization Code Grantの例)



28

4-1. 認証機能 認証画面表示 (Authorization Code Grantの例)

認証画面表示

- エンドポイント：
[https://{ \\$IP }/wso2am/oauth2/authorize](https://{ $IP }/wso2am/oauth2/authorize)
- HTTPメソッド：
GET
- リクエストパラメータ：

認証・認可サーバのIPアドレスについては別途展開します

パラメータ名	説明	例
scope	“default”を指定してください。	default
response_type	Authorization Code Grantの場合は、“code”を指定してください。	code
redirect_uri	アプリケーション登録時に指定したコールバックURLを指定してください。(※)	https://example.com/oauth/callback
client_id	アプリケーション登録時に発行されたConsumer Keyを指定してください。	ejwiao23tr4

(※) 利用者が認証画面で認証した後、redirect_uriで指定したURLへ認可コードが渡されます。この認可コードを「2. アクセストークン取得」で利用します。
例) <https://example.com/oauth/callback?code=abcdef>

認可コード

29

4-1. 認証機能 アクセストークン取得 (Authorization Code Grantの例)

アクセストークン取得

- エンドポイント：
[https://{ \\$IP }/wso2am/oauth2/token](https://{ $IP }/wso2am/oauth2/token)
- HTTPメソッド：
POST
- Content-Type：
application/x-www-form-urlencoded
- リクエストパラメータ：

パラメータ名	説明	例
code	「5-2. 認証画面表示」で取得した認可コードを指定してください。	abcdef
grant_type	Authorization Code Grant の場合は、“authorization_code”を指定してください。	code
client_secret	アプリケーション登録時に発行されたConsumer Secretを指定してください	poiif0we9aijkewahe89
redirect_uri	「5-2. 認証画面表示」と同様	https://example.com/oauth/callback
client_id	「5-2. 認証画面表示」と同様	ejwiao23tr4

30

4-1. 認証機能 レスポンス (Authorization Code Grantの例)

● レスポンスボディ

JSON形式

```
{ "scope": "default", "token_type": "Bearer", "expires_in": 2413, "refresh_token": "e156236ef50596b80d44adbb1c2773b0", "access_token": "4e88f99fa193bafbeb41c528b9b9e070" }
```

式で以下のパラメータが返却されます。

パラメータ名	説明	例
scope	「5-2. 認証画面表示」で指定したscopeの値が返却されます。	default
token_type	"Bearer"が返却されます。	Bearer
expires_in	アクセストークンの有効期限（秒）が返却されます。	2413
refresh_token	リフレッシュトークンが返却されます。あアクセストークンの有効期限延長に使用します（※）	e156236ef50596b80d44adbb1c2773b0
access_token	アクセストークンが返却されます。API呼び出しに使用します。	4e88f99fa193bafbeb41c528b9b9e070

例)

```
{ "scope": "default", "token_type": "Bearer", "expires_in": 2413, "refresh_token": "e156236ef50596b80d44adbb1c2773b0", "access_token": "4e88f99fa193bafbeb41c528b9b9e070" }
```

(※) アクセストークンには有効期限があるため、継続して使用するためにはリフレッシュトークンを使用してアクセストークンの有効期限を延長する必要があります。詳細は下記URLを参照。
<https://docs.wso2.com/display/IS530/Refresh+Token+Grant>

4-1. 認証機能 API呼び出し (Authorization Code Grantの例)

APIの呼び出し

各APIを呼び出す際に、HTTPヘッダに「5-3. アクセストークン取得」で取得したアクセストークンを指定します。

● HTTPヘッダ

ヘッダフィールド	値
Authorization	Bearer 【OAuth 2.0のアクセストークン文字列】

例) コンテキスト管理機能のデータ参照 (queryContext) のAPIを呼び出す場合

```
(curl -k -v -X POST "https://{IP}/orion/v1.0/queryContext" -s -S --header "Authorization: Bearer 4e88f99fa193bafbeb41c528b9b9e070" --header "Content-Type: application/json" --header "Accept: application/json" --header "Fiware-Service: 事業者 A" --header "Fiware-ServicePath: /B部門/C課" -d @- | python -mjson.tool) <<EOF
{
  "entities": [
    {
      "type": "Street",
      "isPattern": "false",
      "id": "Street4"
    }
  ]
}
EOF
```

4-2. 各APIの利用方法

主に利用が見込まれるAPIについて以降に利用方法を記載します。

API名	利用シーン
updateContext	データをコンテキスト管理に登録する
queryContext	コンテキスト管理に登録されているデータを参照する
subscribeContext	履歴データストレージに履歴データを蓄積する
contextEntities	履歴データストレージに格納された履歴情報を参照する

※本資料およびアプリケーション開発ガイドに記載の各APIは、開発するwebアプリ側からHTTPリクエストをPOST/GETメソッドにて送信、HTTPレスポンスを受信して使用する事を想定しております。各事業者で実装してください。

※資料内の各サーバのIPアドレスやサービスのポート番号については別途正式な環境の値を展開いたします。

各APIの詳細については「アプリケーション開発ガイド（データ分析参照編）」、「アプリケーション開発ガイド（データ収集蓄積編）」を参照

4-2. 各APIの利用方法 データ登録(1/5)

データ登録のAPI（updateContext）を利用する

- エンドポイント：
<https://{IP}/orion/v1.0/updateContext>
- HTTPメソッド：
POST
- HTTPヘッダ：
Content-Type : application/json
Accept : application/json
Authorization : Bearer [\\${アクセストークン}](#)
Fiware Service : 事業者A(例)
Fiware ServicePath : /B部門/C課(例)

4-2. 各APIの利用方法 データ登録(2/5)

- リクエストボディ : json形式で下記の内容を指定する。

パラメータ名	型	説明	例
contextElements	Array	登録するコンテキスト情報	
type	String	登録対象のエンティティタイプ	Room
isPattern	String	false (固定)	False
id	String	登録対象のエンティティ	Room-1
attributes	Array	登録対象の属性のリスト。	
name	String	登録する属性名	temperature
type	String	登録する属性のタイプ	float
value	String	登録する属性の値	20.5
updateAction	Object	更新モードの指定 APPEND : 作成・更新 APPEND_STRICT : 作成 (更新不可) UPDATE : 更新 (作成不可) DELETE : 削除 REPLACE : 置換 詳細は Update action types を参照	APPEND

参考 : Update action types

http://fiware-orion.readthedocs.io/en/master/user/update_action_types/

35

4-2. 各APIの利用方法 データ登録(3/5)

https://\${IP}/orion/v1.0/updateContext

- リクエストボディ (例)

```
{
  "contextElements": [
    {
      "type": "Room",
      "isPattern": "false",
      "id": "Room-1",
      "attributes": [
        {
          "name": "temperature",
          "type": "float",
          "value": "20.5"
        },
        {
          "name": "humidity",
          "type": "integer",
          "value": "50"
        }
      ]
    }
  ],
  "updateAction": "APPEND"
}
```

登録するコンテキスト情報の指定。

登録する属性情報の指定。

更新モードの指定。
APPENDを指定しているため、
なければ作成、あれば更新。

36

4-2. 各APIの利用方法 データ登録(4/5)

●レスポンスボディ（例）

```
{
  "contextResponses": [
    {
      "contextElement": {
        "type": " Room ",
        "isPattern": "false",
        "id": " Room-1 ",
        "attributes": [
          {
            "name": "temperature",
            "type": "float",
            "value": ""
          },
          {
            "name": "humidity",
            "type": "integer",
            "value": ""
          }
        ]
      },
      "errorCode": {
        "code": "200",
        "reasonPhrase": "OK"
      }
    }
  ]
}
```

登録するコンテキスト情報が表示される

登録する属性情報が表示される。Valueは空で返却

ContextElement毎のステータスコード

37

4-2. 各APIの利用方法 データ登録(5/5)

■ 禁則文字

- 下記の文字は直接利用できないためデータ登録時に利用する場合はURLエンコードを実施した上で登録してください。（参照時は逆変換）

禁則文字	エンコード後	備考
<	%3C	不等号（より小）（0x3c）
>	%3E	不等号（より大）（0x3e）
“	%22	ダブルクォート（0x22）
'	%27	シングルクォート（0x27）
=	%3D	イコール（0x3d）
;	%3B	セミコロン（0x3b）
(%28	左小括弧（0x28）
)	%29	右小括弧（0x29）
%	%25	パーセント（0x25） ※パーセントは禁則文字ではありませんが、他の変換と競合するため同様にURLエンコードを実施してください

38

4-2. 各APIの利用方法 データ参照(1/5)

データ参照のAPI (queryContext) を利用する

- エンドポイント：
[https://\\${IP}/orion/v1.0/queryContext](https://${IP}/orion/v1.0/queryContext)
- HTTPメソッド：
POST
- HTTPヘッダ：
Content-Type : application/json
Accept : application/json
Authorization : Bearer [\\${アクセストークン}](#)
Fiware Service : 事業者A(例)
Fiware ServicePath : /B部門/C課(例)
- クエリパラメータ：

クエリパラメータを追加することで条件を付けて検索結果を表示する

パラメータ名	説明	例
limit	一回のリクエストで取得するコンテキストの件数上限の指定。デフォルトは20。	20
offset	検索結果の返却する位置。デフォルトは0。	0
details	検索結果の詳細情報を返却する。デフォルトはoff。	off

39

4-2. 各APIの利用方法 データ参照(2/5)

- リクエストボディ : json形式で下記の内容を検索条件として指定する。

パラメータ名	型	説明	例
entities	Array	取得対象のリスト	
type	String	取得対象のエンティティタイプ	Room
isPattern	String	idに正規表現を使用するか否か "true"か"false"を指定する	true
id	String	取得対象のエンティティID。isPatternがtrueの場合は正規表現で指定可能。 詳細は Using regular expressions in payloads を参照	.*
attributes	Array	取得対象の属性のリスト。省略時は全ての属性を返却する。	temperature
restriction	Object	フィルタリング条件。属性や位置情報でのフィルタリングが可能。 詳細は NGSIv1 Filtering を参照	

参考 : FIWARE NGSI APIv1 Walkthrough

http://fiware-orion.readthedocs.io/en/master/user/walkthrough_apiv1/index.html

参考 : Using regular expressions in payloads

http://fiware-orion.readthedocs.io/en/master/user/regex_in_payload/index.html

参考 : NGSIv1 Filtering

<http://fiware-orion.readthedocs.io/en/master/user/filtering/#ngsiv1-filtering>

40

4-2. 各APIの利用方法 データ参照(3/5)

https://\${IP}/orion/v1.0/queryContext?limit=100&offset=200&details=on

● リクエストボディ (例)

```

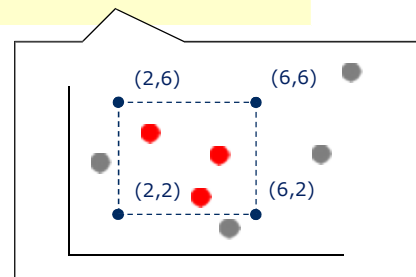
{
  "entities": [
    {
      "type": "Room",
      "isPattern": "true",
      "id": ".*"
    }
  ],
  "attributes": [
    "temperature"
  ],
  "restriction": {
    "scopes": [
      {
        "type": "FIWARE::Location",
        "value": {
          "polygon": {
            "vertices": [
              {"latitude": "2", "longitude": "2"},
              {"latitude": "2", "longitude": "6"},
              {"latitude": "6", "longitude": "6"},
              {"latitude": "6", "longitude": "2"}
            ]
          }
        }
      }
    ]
  }
}

```

エンティティ検索条件の指定。
Typeが"Room"、の任意のIDの
データを取得する例。

取得対象の属性絞り込みの指定。
"temperature"の属性のみを取
得する例。

位置情報での絞り込みの指定。
ポリゴン指定した範囲の位置情報
をもつデータのみを取得する例



クエリパラメータ
取得位置の指定。
100件ずつ取得し、3ページ目
を取得する例。

4-2. 各APIの利用方法 データ参照(4/5)

● レスポンスボディ：json形式で下記のコンテキスト情報が返却される。

パラメータ名	型	説明	例
contextResponses	Array	検索結果のリスト	
contextElements	Objet	コンテキスト情報のオブジェクト	
type	String	エンティティタイプ	Room
isPattern	String	False (固定)	false
id	String	エンティティID	Room-1
attributes	Array	コンテキストの属性情報	
name	String	属性名	temperature
type	String	属性の型	float
value	String	属性の値	20.5
statusCode	Object	ステータス情報	
code	String	ステータスコード	200
reasonPhrase	String	メッセージ	OK
errorCode	Object	エラー情報。エラーまたはdetails=onの場合に出力される。	
code	String	エラーコード	
reasonPhrase	String	メッセージ	
details	String	詳細情報	

4-2. 各APIの利用方法 データ参照(5/5)

●レスポンスボディ (例)

```
{
  "contextResponses": [
    {
      "contextElement": {
        "attributes": [
          {
            "name": "temperature",
            "type": "float",
            "value": "20.5"
          }
        ],
        "id": "Room-1",
        "isPattern": "false",
        "type": "Room"
      },
      "statusCode": {
        "code": "200",
        "reasonPhrase": "OK",
      }
    },
    {
      ...
    }
  ],
  "errorCode": {
    "code": "200",
    "reasonPhrase": "OK",
    "details": "Count: 322"
  }
}
```

検索条件に合致したコンテキスト情報が返却される。出力されるのはlimitで指定した件数が上限。

検索条件で属性を指定した場合は、指定した属性のみが出力される。

details=onを指定することで詳細情報が出力される。
検索条件に合致する全件数情報はここに出力されるため、これを利用してページネーションを実施する。

43

4-2. 各APIの利用方法 データの履歴情報の蓄積(1/4)

■ データ更新通知のAPI (subscribeContext) を利用する

- データの履歴を残すには、更新通知の設定が必要となる。蓄積履歴の通知先(reference)に履歴データストレージを指定する。
- エンドポイント：
<https://{IP}/orion/v1.0/subscribeContext>
- HTTPメソッド：
POST
- HTTPヘッダ：
Content-Type : application/json
Accept : application/json
Authorization : Bearer [\\${アクセストークン}](#)
Fiware Service : 事業者A(例)
Fiware ServicePath : /B部門/C課(例)

参考 : Context subscriptions

https://fiware-orion.readthedocs.io/en/master/user/walkthrough_apiv1/index.html#context-subscriptions

44

4-2. 各APIの利用方法 データの履歴情報の蓄積(2/4)

- リクエストボディ : json形式で下記の内容を検索条件として指定する。

パラメータ名	型	説明	例
entities	Array	取得対象のリスト	
type	String	取得対象のエンティティタイプ	Room
isPattern	String	idに正規表現を使用するか否か "true"か"false"を指定する	false
id	String	取得対象のエンティティID。isPatternがtrueの場合は正規表現で指定可能。	Room-1
attributes	Array	取得対象の属性のリスト。省略時は全ての属性を返却する。	temperature
reference	String	データ更新通知を送信する通知先を指定。 履歴データストレージに格納するため以下を指定する。 http://cygnus-comet:5050/notify	
duration	String	有効期限をISO8601標準フォーマットの期間仕様で指定（省略時24時間）	P1M
notifyConditions	String	通知条件	
type	String	"ONCHANGE"（固定）	ONCHANGE
condValues	Array	監視対象となる属性名称のリスト。省略時、すべての属性が対象となる	temperature
throttling	String	最小通知間隔をISO8601標準フォーマットの期間仕様で指定。省略可能。	PT5S

45

4-2. 各APIの利用方法 データの履歴情報の蓄積(3/4)

https://\${IP}/orion/v1.0/subscribeContext

- リクエストボディ（例）

```
{
  "entities": [
    {
      "type": "Room",
      "isPattern": "false",
      "id": "Room-1"
    }
  ],
  "attributes": [
    "temperature"
  ],
  "reference": "http://cygnus-comet:5050/notify",
  "duration": "P1M",
  "notifyConditions": [
    {
      "type": "ONCHANGE",
      "condValues": [
        "temperature"
      ]
    }
  ],
  "throttling": "PT5S"
} EOF
```

エンティティ検索条件の指定。
Typeが"Room"の指定したID
「Room-1」のデータを取得する例。

取得対象の属性 絞り込みの指定。
"temperature"の属性のみを取得する例。

reference : 履歴情報として格納するため履歴
データストレージを指定する

有効期限を1か月に指定

監視対象を温度に設定、更新があるとデータが蓄積される

5秒間隔で確認

46

4-2. 各APIの利用方法 データの履歴情報の蓄積(4/4)

- レスポンスボディ：json形式で下記のコンテキスト情報が返却される。

パラメータ名	型	説明	例
subscribeResponse	Array	サブスクライブレスポンス	
duration	String	設定された有効期限	P1M
subscriptionId	String	データ更新通知予約の識別子。データ更新通知予約の更新/削除時に使用。	58d76733・・・
throttling	String	設定された最小通知間隔。	PT5S

- レスポンスボディ（例）

```
{
  "subscribeResponse": {
    "duration": "P1M",
    "subscriptionId": "58d76733e46f5af904162a01",
    "throttling": "PT5S"
  }
}
```

サブスクリプションIDが取得できていること

47

4-2. 各APIの利用方法 履歴データを参照する(1/3)

履歴データ参照のAPI（contextEntities）を利用する

- エンドポイント：
<https://{IP}/comet/v1.0/contextEntities>
- HTTPメソッド：
GET
- HTTPヘッダ：
Content-Type : application/json
Accept : application/json
Authorization : Bearer [\\${アクセストークン}](#)
Fiware Service : 事業者A(例)
Fiware ServicePath : /B部門/C課(例)

48

4-2. 各APIの利用方法 履歴データを参照する(2/3)

● リクエストボディ (例)

```
curl -s -X GET
"https://${IP}/comet/v1.0/contextEntities/type/car/id/car1/attributes/speed?
hLimit=50&hOffset=0&dateFrom=2017-01-01T00:00:00Z&dateTo=2017-12-
31T23:59:59Z" ¥
--header "Content-Type: application/json" ¥
--header "Fiware-Service: 事業者A" ¥
--header "Fiware-ServicePath: /xxxx/事業者A/" ¥
--header "Accept: application/json" ¥
--header "Authorization: Bearer ${TOKEN}" ¥
| python -mjson.tool
```

レスポンスに含まれる履歴数やオフセット
履歴の範囲 (時間) を指定

参照したいContext Element (Context
Entity) のtype、id、attributeを指定。

49

4-2. 各APIの利用方法 履歴データを参照する(3/3)

● レスポンスボディ (例)

```
{
  "contextResponses": [
    {
      "contextElement": {
        "attributes": [
          {
            "name": "speed",
            "values": [
              {
                "attrType": "integer",
                "attrValue": "40",
                "recvTime": "2017-03-25T09:43:47.011Z"
              },
              {
                "attrType": "integer",
                "attrValue": "50",
                "recvTime": "2017-03-25T09:43:48.151Z"
              }
            ]
          }
        ],
        "id": "car1",
        "isPattern": false,
        "type": "car"
      },
      "statusCode": {
        "code": "200",
        "reasonPhrase": "OK"
      }
    }
  ]
}
```

履歴データストレー
ジから車の速度デー
タを取得

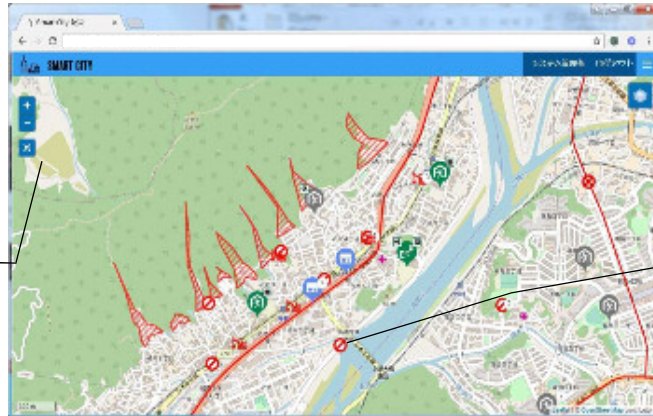
結果コード

50

4-3. 地図データの利用方法(1/4)

GISの地理情報を利用してアプリケーション上に地図とデータを合わせた直観的な可視化が可能

アプリ上で地図データと共に収集した情報をマッピングすることで可視性を向上。GeoJSON形式のデータ併用する事で図示したようなポイント、ライン、ポリゴン(範囲)の表記も可能となる。



利便性

既存ライブラリを活用することで、地図表示に必要な拡大縮小といった機能を容易に実装可能

可視性

収集した情報を地図上にマッピングすることで、地域全体の情報の可視性向上

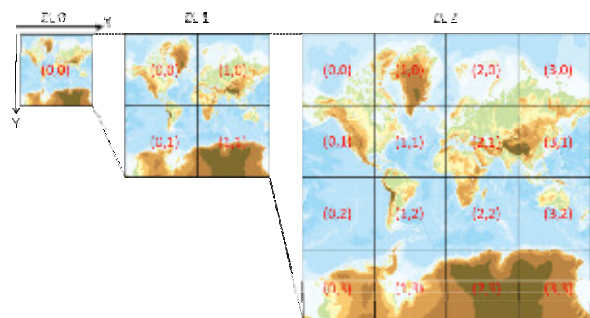
詳細は、『データ活用基盤サービス (FIWARE) アプリケーション開発ガイド (GIS編)』を参照

51

4-3. 地図データの利用方法(2/4)

ズームレベルとタイルの座標データを指定し地図画像のデータを取得

GISと連動しベース地図(例：世界地図)となる画像をタイル(Tile Map Service)方式で取得する事が可能。ズームレベルに合わせてベース画像を分割し、それぞれに座標を割り当てることで指定した地図画像ファイル(png形式)を取得しアプリ上で利用する事が可能。



GISサーバのIP、ポートについては別途展開します

【URL】

[http://\\$\[{\\$IP}:{\\$PORT}\]/](http://$[{$IP}:{$PORT}]/)

【コマンド例】

`gis/geoserver/gwc/service/tms/1.0.0/mn2gt@EPSG:900913@png/{z}/{x}/{y}.png`

ここで、{x}、{y}、{z}の意味は以下のとおりです。

{x}：タイル座標のX値

{y}：タイル座標のY値

{z}：ズームレベル

52

4-3. 地図データの利用方法(3/4)

例として地図表示のサンプルコードを以下に記載します (Leaflet.jsを使用)

```
<html>
<head>
<meta http-equiv="X-UA-Compatible" content="IE=EmulateIE10">
<link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.css"/>
<script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"></script>
<script type="text/javascript">

function init () {

// GIS Data Providerレイヤの定義
var gisdataproviderLayer = L.tileLayer(
'http://{IP}/gis/geoserver/gwc/service/tms/1.0.0/mn2gt@EPSG:900913@png/{z}/{x}/{y}.png', {
tms: true,
attribution: '&copy; <a href="http://osm.org/copyright" target="_blank">OpenStreetMap</a> contributors'
}
);
// 地図の初期表示設定
mymap = L.map(
'mapid', {
center: [35.571841, 139.66577],
zoom: 16,
layers: [gisdataproviderLayer]
}
);
</script>
</head>
<body onload="init()">
<div id="mapid" style="width:100%; height:100%;"></div>
</body>
</html>
```

Leaflet.jsの読み込み

レイヤの定義
・本システムのGISを指定
・OSMのコピーライトを指定

地図の表示処理
・座標
・ズームレベル

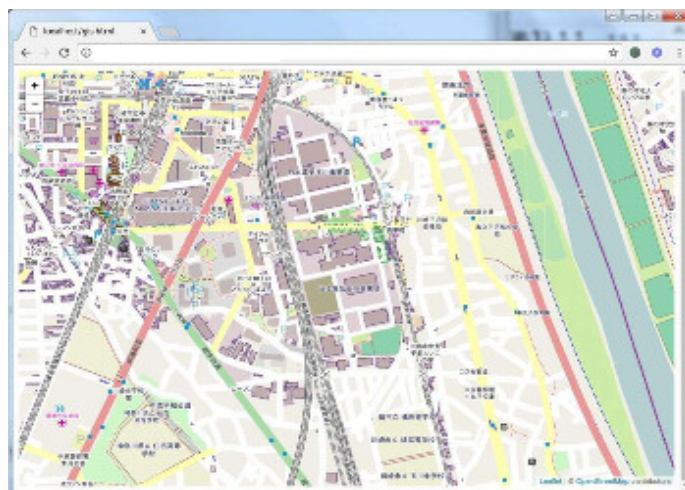
地図の表示エリア

Leaflet.jsのその他機能は下記を参照
<http://leafletjs.com/>

53

4-3. 地図データの利用方法(4/4)

サンプルコードの出力結果



注意事項

データ利活用基盤サービスは、OpenStreetMapの地図を利用しています。本機能を利用する際はライセンス上OpenStreetMapのクレジット表記を行う必要があります。

参考：<https://www.openstreetmap.org/copyright>

54