屋内測位新宿ターミナル
オープンデータ仕様書

令和２年３月改訂版

目次

1. はじめに

東京２０２０オリンピック・パラリンピック競技大会開催時には、多くの人々が東京を訪れることが見込まれており、駅利用者が安全かつ円滑に移動できる情報環境を整備する必要があり、既に新宿駅などでは案内サインの表現の統一化に向けて取り組みを行っています。現在東京都では、新宿駅における案内サインを補足するため、屋内測位技術を活用した案内誘導サービスを検討しより円滑な移動をサポートする環境の整備を進めています。

はじめて新宿駅に来た旅行者やベビーカー及び車いす利用者など、様々な属性の来訪者のスムーズな移動を実現できる民間アプリの開発促進を目指し、２０１９年１１月に東京都主導のもと実施した、屋内測位技術を活用した案内誘導サービスの実証デモ結果を取りまとめ、新宿駅の乗り換えルートに関するデータをオープンデータとして公開します。

本仕様書は、本案件で活用した地下街等の公共的屋内空間を主対象とし（ただし、対象となる施設の敷地内の屋外空間も含む)、階層別の地理空間情報データおよびナビゲーション情報に関わる地図データの仕様について規定するものです。

2. 基本方針

2.1. 基本方針

「令和元年度 ターミナル駅における屋内測位情報を活用した案内誘導に関する検討業務委託」案件において、案内ルート表示に必要な情報として活用した情報を提供するものとします。

2.2. 格納情報

2.2.1. 施設情報

以下の情報を提供するものとする。

・トイレやエレベーター等の緯度経度
・施設名称
・施設の矩形情報

2.2.2. ナビゲーション情報

以下の情報を提供するものとする。

・結節点情報
・結節点間を結ぶ線情報
・結節点と結節点の紐付け情報

3. GeoJSON ファイルデータ仕様

3.1. 概要

本データは GeoJSON 形式で指定されたジオメトリデータを含む HERE Venues Community の配布可能なデータファイルです。
GeoJSON 形式の仕様は http://www.geojson.org を参照ください。

データセットには 3 つの異なるファイルタイプがあります。

・コミュニティマップ

マップ対象施設の「コミュニティ情報、図面情報、レベル情報」が含まれています。このファイルではコミュニティ（施設）の基本情報を提供します。ジオメトリ（区画）の情報は別のファイルにて提供のため、このファイルにはジオメトリ情報は含まれません。このファイルの構造は、GeoJSON の構造に類似した形式になっています。

・レベルジオメトリ

このファイルには、特定のレベルのすべてのジオメトリ情報が含まれています。このファイルは、GeoJSON Feature Collection の形式です。

・コミュニティエンティティ

このファイルには、コミュニティ内のすべてのエンティティ情報が含まれます。ファイルの構造は、GeoJSON の構造に類似した形式になっています。

3.1.1. 曲線について

HERE Venues マップジオメトリには、SVG に似たベジェ曲線が含まれています。これらの曲線は、標準の GeoJSON ジオメトリタイプを提供するために平坦化されています。さらに、ジオメトリファイルは平坦化せずに使用できます。このファイルには、このドキュメントで定義されているカスタム GeoJSON ジオメトリタイプ「Path」が含まれています。

3.1.2. 言語について

コミュニティ情報に複数の言語のデータがある場合、データファイルには全て言語のデータセットを格納しています。

・コミュニティオブジェクトでは、デフォルトの言語とすべての言語のリストが提供されます。
・オブジェクトの「プロパティ」エントリには、言語非依存またはデフォルト言語に対応するすべてのプロパティが含まれています。

・追加の言語ごとに、プロパティエントリが追加されます。これらのエントリのラベルは、「プロパティ」+「:」+言語コードです。これらのエントリには、言語ごとに属するプロパティのみ情報が含まれています。

3.1.3. プロパティについて

各オブジェクトタイプのプロパティの説明には、許可されるプロパティがリストされています。これには、HERE Venues 標準オブジェクトプロパティのエントリが含まれます。これらは、プロパティプリセットファイルにあります。GeoJSON ファイルには、HERE Venues マップオブジェクトの標準プロパティの他に追加のプロパティがあるかもしれません。これらの GeoJSON プロパティについても、このドキュメントにリストされています。

標準の HERE Venues プロパティは、通常は文字列値ですが、文字列の配列である値を持つこともできます。各プロパティの推奨形式は、プリセットドキュメントに記載されています。

プロパティの配列に複数の値がある場合、GeoJSON のプロパティには文字列の JSON 配列が指定されます。それ以外の場合、文字列値が与えられます。

3.1.4. 内部アドレスについて

内部アドレスは、すべてが内部アドレス文字列として表示されます。各キーと値のペアは、キー+スペース+値として表示されます。キーが null の場合、値のみが表示されます。複数のキーと値のペアを持つジオメトリとエンティティの両方について、キーと値のペアはカンマ+スペース「, 」で区切られます（表記例　⇒　建物 3, 部屋 241)。

3.2. コミュニティマップ

コミュニティマップファイルには、コミュニティの組織構造が含まれています。コミュニティマップ JSON とそれに含まれるオブジェクトで構成されます。

名前の形式：com-map_［コミュニティ ID］_［データバージョン］. json

データバージョンは、リリースプロセス中のスナップショットから取得されます。リリースされたスナップショットではなくアクティブ DB から直接描画されたデータの場合、バージョンは 0 として指定されます。

3.2.1. Community Map JSON

■Community Map JSONの書式

{

"id"：コミュニティ ID、

"obj_type"： オブジェクトタイプ："CommunityMap"、

"map_version"：マップバージョン、

"entity_version"：エンティティバージョン、

"properties"：プロパティマップ//以下を参照

"languages"：言語配列

"default_lang"： 言語、

"location"：コミュニティの lat lon geojsonポイント

"drawings"：[描画リスト...] //「3.2.2. JSONの描画」を参照

}

■コミュニティマップ JSONのプロパティ

HERE Venuesメタデータのプロパティキーで定義された標準コミュニティプロパティ

3.2.2. Drawing JSON

■DrawingJSONの書式

{

"id"：図面 ID、

"obj_type"：オブジェクトタイプ："Drawing"、

"levels"：レベルリスト、//「3.2.3. レベル JSON」を参照

"properties"：プロパティマップ、//以下を参照

"object_groups"：マップタイプリスト、//以下を参照

"ref_frame"：{

"transform"：ローカル座標から緯度経度座標への変換

"height"：ローカル座標の高さ

"width"：ローカル座標の幅

"angle_deg"：デフォルトの角度、

"local2m"：変換するスケール係数メートル単位のローカル座標

}

}

■Drawing JSON のプロパティ

・標準で提供される図面プロパティは HERE Venues メタデータのプロパティキーで定義されいます。

・is_root（オプション、デフォルトは false）-このフラグは、これがコミュニティのルート図面であるかどうかを示します。

・非推奨:display_name-これは、図面の表示名です。図面には、display_name とは別に「name」プロパティというデータが存在します。

「display_name」は、図面プロパティに表示されていますが、将来的には削除されるかもしれません。その場合には代わりにプロパティ「name」を使用してください。

■ローカル座標

ローカル座標は、会場に対してローカルに定義された座標のセットです。座標系は、北向きではなく自然な向きで定義されます。ローカル座標は、Path json を持つファイルのバージョンで使用されます。標準の GeoJSON ジオメトリタイプのファイルは、標準の WGS84 座標を使用します。

3.2.3. Level JSON

■Level JSON の書式

{

"id"：レベル ID、

"obj_type"：オブジェクトタイプ："Level"、

"properties"：プロパティマップ、//下記参照

}

■Level JSON のプロパティ

・HERE Venues メタデータのプロパティキーで定義された標準レベルのプロパティ。

・zlevel - レベルの zlevel、または順序付けパラメーター

・main（オプション、デフォルトは false）-これは、この図面がこの図面のメインレベルであるかどうかを示します。

・root_geom - ルート GeoJSON のジオメトリ ID

・parent_level - 親レベルのレベル ID

3.2.4. Level Geometry JSON

Level Geometry JSON には 2 つのバージョンがあり、1 つは標準のフラット化された GeoJSON を使用し、もう 1 つは Path GeoJSON を使用します。

標準の GeoJSON 名の形式：

geojson-level-geom_［コミュニティ ID]-[図面 ID]-[レベル ID］ _［データバージョン］.json

Path GeoJSON 名の形式：

path-level-geom_［コミュニティ ID]-[図面 ID]-[レベル ID］_［データバージョン］.json

データバージョンは、リリースプロセス中のスナップショットから取得されます。リリースされたスナップショットではなくアクティブ DB から直接描画されたデータの場合、バージョンは 0 として指定されます。

■LevelGeometry 機能コレクション GeoJSON

{

"id"：レベル ID、、

"drawing_id"：図面 ID、

"community_id"：コミュニティ ID、

"map_version"：マップバージョン、

"entity_version"：エンティティバージョン、

"obj_type"：レベルジオメトリ、

"type"：機能コレクション、

" properties "：プロパティマップ、// Level オブジェクト

" crs "：crs と同様//以下を参照

" features "：ジオメトリリスト、//「3.2.5. Geometry Feature GeoJSON」を参照

}

■CRS

GeoJSON のオプションパラメーターです。CRS が存在しない場合、データは WGS84 緯度経度座標で提供されます。CRS が null の場合、座標は推定できません。

HERE Venues Maps データの場合、データは、標準 GeoJSON の WGS84 緯度および経度座標、または Path GeoJSON の描画オブジェクトで定義されたローカルマップ座標のいずれかで提供されます。緯度と経度を使用する場合、CRS は省略されます。ローカル座標が使用される場合、CRS エントリは null の値で含まれます。

3.2.5. Geometry Feature GeoJSON

geometry geojson には 2 つの形式があります。それらは、平坦化されたジオメトリパスを含む標準の GeoJSON です。次に、フル曲線パスを表示する非標準の Path GeoJSON があります。

■Geometry Feature GeoJSON の書式

```
{
    "id"：ジオメトリ ID、
    "obj_type"：ジオメトリ、
    "type"：特徴、
    "properties"：プロパティマップ、    //下記を参照
    "group"：グループ ID、              //レベル間で geom をグループ化するために使用
    "is_group_main"：メインの geom、    //下記を参照
    "geometry"：geometry GeoJSON またはジオメトリパス、
    "line_width"：物理的な幅、          //線形領域のみ下記を参照
    "label_area"：ラベル領域、          //下記を参照
    "location"：ポイント geoJSON        //非ポイントジオメトリのみ
    "links"：リンク json                //下記を参照
}
```

■Geometry Feature GeoJSON のプロパティ

・HERE Venues メタデータのプロパティキーで定義された標準のジオメトリプロパティ。
・is_root（オプション、デフォルトは false）-図面用のルート geom
・int_address（オプション、デフォルトは none）-該当する場合、「room 213」などの内部アドレス文字列。
・entity_name-該当する場合、ジオメトリに関連付けられたエンティティから派生した名前。ジオメトリには、独自の「名前」プロパティもあります。
・entities-該当する場合、このジオメトリに関連付けられた標準エンティティのリスト。
・internal_entities-該当する場合、このジオメトリに関連付けられた内部エンティティのリスト。

エンティティリストと内部エンティティリストの大まかな違いは、ジオメトリのエンティティ名がエンティティリストのオブジェクトから派生していることです。内部エンティティはジオメトリに含まれていますが、オブジェクトの標準エンティティ名の作成には使用されません。内部エンティティを使用できる例は、個々の店舗がセクションごとにリストされていない区割りのモールマップです。

3.2.6. Root Geometry

レベルにはルートジオメトリが含まれる場合があります。このジオメトリは、別の図面である親図面の子（通常）ジオメトリです。

ルートジオメトリには3つの表示シナリオがあります。

1）通常の表示-この場合、ジオメトリは親図面で表示されるのと同じ方法で子図面に表示されます。親と子レベルの両方のジオメトリオブジェクトに同じ情報が含まれます。

2）表示なし-この場合、ルートジオメトリは子に表示されません。ここでは、子レベルのジオメトリには含まれません。

3）背景として表示-この場合、ジオメトリオブジェクトを特別な「背景」として表示する必要があります。ここでは、ジオメトリが子レベルのジオメトリオブジェクトに含まれ、プロパティのセットが削減されています。

・type - バックグラウンドタイプ名として指定されます。

・is_root-trueに設定

■線幅

基本的なジオメトリタイプは、ポイント、ライン、エリアです。HERE Venuesでは、追加するタイプ線形エリアを定義しています。これは、物理的な幅に対応する追加のパラメーター幅を持つ線として定義されるエリアオブジェクトです。この幅は、座標系がWGS84の場合はメートル単位で指定され、ローカル座標系が使用される場合はローカル座標単位で指定されます。

■ジオメトリグループ

ジオメトリグループは、複数の階層レベルにわたって存在するジオメトリをグループ化するために使用されます。グループには、「is_group_main」フラグで示されるメインジオメトリがあります。メインジオメトリのジオメトリIDは、特定のグループの単一のジオメトリIDを表すためのスタンドとしてしばしば使用できます。「is_group_main」フラグはオプションであり、存在しない場合はデフォルトでfalseになります。

■ラベルエリア

これは、ラベルをレンダリングするために作成された長方形です。ジオメトリに「external_label」プロパティがない限り、長方形はジオメトリで完全に囲まれます。値はjson配列として与えられます：[中心X、中心Y、幅、高さ、角度]。

・中心のXとYは、ジオメトリと同じ座標系で指定されます。

・ローカル座標系が使用される場合、幅と高さは WGS84 およびローカル座標のメートル単位で指定されます。

3.2.7. Path GeoJSON

GeoJSON 構造は標準で定義されています。
Path GeoJson は次のように定義されます。

```
{
"type" : Path、
     "coordinates" : [[整数コマンド、x0、y0 ...]、...]
     "area" : true / false
 }
```

座標は命令の配列として与えられます。各命令には、次の表に示すコマンド整数と、コマンドの値に依存する引数のリストが含まれます。コマンドは、SVG パスコマンドから派生しています。ただし、JSON は文字をサポートせず、パスをよりコンパクトにするため、整数を使用してコマンドを表現します。SVG コマンド値から JSON コマンド整数へのマッピングを以下に示します。

SVG コマンドのリファレンスについては、http://www.w3.org/TR/SVG/paths.html を参照してください。

※注：SVG コマンドから JSON コマンドへのマッピングの起源は、多くの場合そうであるように、レガシー要件から来ています。相対コマンドのインデックスが絶対コマンドよりも 10 大きいという事実は、厳密には現在 10 個のコマンドがあるという事実の偶然です。新しい SVG コマンドが追加されると、この特定の関係は保持されなくなります。

SVG Cmd	JSON Cmd	名	Arg Cnt	サポート
M	0	～へ移動	2	Y
m	10	移動、相対	2	
L	1	行先	2	Y
l	11	行先、相対	2	
H	6	水平線	1	
h	15	水平線の終点、相対	1	
V	6	縦線	1	
v	16	垂直線の終点、相対	1	

Q	2	二次曲線へ	4	Y
q	12	二次曲線の終点、相対	4	
T	7	滑らかな 2 次曲線	2	
t	17	滑らかな 2 次曲線、相対	2	
C	3	立方体曲線	6	Y
c	13	3 次曲線の相対、	6	
S	8	滑らかな 3 次曲線	4	
s	18	滑らかな 3 次曲線、相対	4	
A	9	Arc へ	7	
a	19	Arc へ、相対	7	
Z	4	Path を閉じる	0	Y
z	14	Path を閉じる	0	

パスの例：

SVG：<path d = "M100,200 C100,100 250,100 250,200 L400,300 Z" />

JSON：[[0,100,200]、[3,100,100,250,100,250,200]、[1,400,300]、[4]]

3.2.8. Community Entities

これらは JSON 形式です。Community entity entry と community map entry とでは、プロパティ情報によりエンティティか、マップのどちらかに属するかを分けることができます。

名前の形式：

com-entity_［コミュニティ ID］_［データバージョン］.json

データバージョンは、リリースプロセス中のスナップショットから取得されます。リリースされたスナップショットではなくアクティブ DB から直接描画されたデータの場合、バージョンは 0 として指定されます。

3.2.9. Community Entities JSON

{

"id"：コミュニティ ID、

"name"：コミュニティ名称、　　//名称のみ、デフォルトの言語で

"obj_type"：コミュニティエンティティ、

"map_version"：地図バージョン、

"entity_version"：エンティティバージョン、

"entities"：エンティティリスト、　　//「3.2.10. Entity JSON」を参照

}

3.2.10.　　Entity JSON

■Entity JSON の書式

{

"id"：エンティティ ID、

"obj_type"：エンティティ、

"properties"：プロパティマップ、　//下記を参照

}

■Entity JSON のプロパティ

・HERE Venues メタデータのプロパティキーで定義された標準エンティティプロパティ

・chain（オプション、デフォルトは null）-チェーン ID、//該当する場合

・int_address（オプション、デフォルトは null）-完全なアドレス文字列。概要の内部アドレスのセクションを参照してください

・geometry（オプション、デフォルトは null）-配列の配列としての一致したジオメトリ。内部配列の形式は次のとおりです。

⇒[geometry ID、isIdentifier]-「isIdentifier」が true の場合、ジオメトリの「エンティティ」リストにジオメトリが表示され、「isIdentifier」が false の場合、ジオメトリの「内部エンティティ」リストに表示されます。

3.3. QA

Q1：GeoJSON ファイルでは、座標に使用される単位は何ですか？

A1：座標は度（緯度、経度）です。距離はメートル単位です。

Q2：GeoJSON ファイルには屋外のプロパティ情報も含まれていますか？

A2：はい、駐車場、近くの道路、駐車スペース、歩道およびその他の詳細などの区画およびプロパティ情報がマップに含まれている場合、その情報は GeoJSON ファイルで利用できます。

4. ナビゲーションファイルデータ仕様

4.1. 概要

HERE Venues ナビゲーションネットワークは、一連のノードとリンクで構成されています。

・ノード - ネットワーク内の許可された場所
・リンク - ノード間のルーティング可能な接続

リンクはネットワーク内の通過可能なパスを形成し、ノード間の単純な接続です。

ナビゲーションノードは単一の物理レベルにあり、ノードの 3 次元座標を形成します。ただし、異なるマップ図面が同じ物理空間を表すことがあるため、単一のノードが複数のマップレベルに表示される場合があります。たとえば空港には、地上レベルのすべてのターミナルのメイン図面（親図面）がある場合があります。さらに、個々のターミナルの図面（子図面）があります。これは、メインの空港マップよりも詳細に表示され、複数の屋内レベルを含みます。ターミナルマップの地上レベルは、メインの空港マップと同じ物理レベルを示します。

子図面に詳細が表示されるジオメトリと同様に、ナビゲーションネットワークでも同じことが言えます。子図面には多くのナビゲーションノードが表示されますが、それらは親図面から省略されています。ナビゲーションデータファイルのノードには、表示するレベルのリストが含まれています。

ナビゲーションノードネットワークは、コミュニティ全体に完全なナビゲーションネットワークを提供する単一の JSON ファイルです。

4.2. ファイル形式

このファイルには、JSON 形式とバイナリ形式の 2 つの形式があります。バイナリ形式は JAVA DataOutputStream で記述され、エンコードされた UTF 文字列など、それによって使用されるデータパターンが含まれます。

4.3. Nav Community Object

■Nav Community Object JSON の書式

```
{
    "id"：コミュニティ ID、
    "v"：コミュニティマップバージョン、
    "ft"： "navnet5"、

    //コミュニティプロパティのリスト
    "p"：{
        プロパティ形式を参照
     }、

    "d"：[nav drawing list ...]、        // レベルを含む図面のリスト
    "n"：[nav node list ...]、           // ナビゲーションノードのリスト
    "l"：[nav link list ...]、           // ナビゲーションリンクのリスト
}
```

■Nav Community Object JSON のプロパティ

・ファイル形式 (UTF 文字列) - “v5”
・コミュニティ ID (int)
・マップバージョン (int)
・コミュニティプロパティ - プロパティリスト形式を参照
・Drawings
  ⇒図面カウント (int)
  ⇒図面 - Nav Drawing Object を参照
・ノード
  ⇒ノード数 (int)
  ⇒ノード - Nav Node Object を参照
・Links
  ⇒リンク数 (int)
  ⇒リンク - -Nav Link Object を参照

補足：HERE Venues データベースでは、コミュニティオブジェクトは、コミュニティマップとコミュニティエンティティと呼ばれる 2 つの部分に分かれています。ナビゲーションファイルには、コミュニティマッププロパティのみが表示されます。コミュニティエンティティのプロパティは省略されています。

4.4. Nav Drawing Object

■Nav Drawing Object JSON の書式

```
{
    "id" : 図面 ID、
    "t": [txx, tyx, txy, tyy, tx0, ty0],  // メルカトル座標からマップ座標への変換
    "p" : {                               // 描画プロパティ
        プロパティの形式を参照
    }、

    "l" : [nav level list ...]                    //この図面のレベルのリスト
}
```

■Nav Drawing Object JSON のプロパティ

・図面 ID (int)
・変換 (6 * double) -メルカトルから自然 (マップ) 座標へのアフィン変換。
・図面プロパティ-プロパティリスト形式を参照
・レベル数 (int)
・レベルリスト - Nav Level Object format を参照

補足：図面オブジェクトに関連付けられた標準の図面プロパティに加えて、派生プロパティ「display_name」が追加されています。これは、図面の推奨タイトルです。

4.5. Nav Level Object

■Nav Level Object JSON の書式

```
{
    "id":レベル ID,
    "z": zlevel 値,
    "p":                                          //レベルプロパティのリスト
    {
        プロパティの形式を参照
    },
}
```

■Nav Level Object JSON のプロパティ

・レベル ID (int)
・Zlevel (int)

・レベルプロパティ - プロパティリスト形式を参照

4.6. Nav Node

■Nav Node JSON の書式

```
"id":ノード ID,
"mx":ノード X 座標,                    //メルカトル座標内
"my":ノード y 座標                     //メルカトル座標内
"p":{                                  //ノードプロパティ
プロパティの形式を参照
},

"l": [list of nav node - レベルオブジェクト...]     //このノードのレベル
```

■Nav Node JSON のプロパティ

・ノード ID (int)
・メルカトル X 座標 (ダブル)
・メルカトル Y 座標 (ダブル)
・レベル数 (バイト)
・レベルリスト - ナビゲーションノードレベルオブジェクト形式を参照してください。
・ノードプロパティ - プロパティリスト形式を参照

4.7. Nav Node-Level Object

■Nav Node-Level JSON の書式

```
{
    "lid": レベル ID,
    "gid": geom id                     //ジオメトリが null でない場合
}
```

■Nav Node-Level JSON のプロパティ

・レベル ID (int)
・Geom ID (int) -このレベルのこのノードを含むジオメトリ。ジオメトリがない場合、値 0 が使用されます。

4.8. Nav Link Object

■Nav Link JSON の書式

```
{
    "id": リンク ID,
    "n1": 開始ノード ID,
    "n2": 終了ノード ID,
    "d": 方向（バイト）,
    "dm": メートル単位の距離,
    "dz":デルタ zlevel,                    // 0 でない場合のみ
    "p":                                   //リンクプロパティのリスト
    {
        プロパティの形式を参照
    },
}
```

■Nav Link JSON のプロパティ

・リンク ID（int）
・開始ノード ID（int）
・終了ノード ID（int）
・方向（バイト）
  1 =フォワード
  2 =逆
  3 =両方
・メートル単位の距離（フロート）
・デルタ zlevel（int）
・リンクプロパティ - プロパティリスト形式を参照

4.9. Property List Format

プロパティはキーと値のペアですが、それぞれに関連付けられた追加情報がいくつかあります。

Name - これがキーです
Value - これは単一の値です
Order - 配列値のプロパティの場合、これはプロパティの順序を示します
Lang - 言語固有の値の場合、言語またはロケールを提供し、そうでない場合は null です

■Property List Format の書式

″p″: [

　[value, name, order, lang(lang が null でない場合)],

　[value, name, order(lang が null である場合)],

　...

]

■Property List Format のプロパティ

プロパティ数（バイト）

プロパティのリスト

　　プロパティ名（UTF）

　　プロパティ値（UTF）

　　注文（int）

　　Lang（UTF）-lang が null の場合、空の文字列がここに配置されます。