Adding Indoor Maps to Your App and Website

Stephane Godbillon, Maps Engineer
Mithilesh Kumar, Maps Engineer
Indoor Mapping Data Format (IMDF)
Indoor Maps Program
Displaying Indoor Maps in an iOS App
Displaying Indoor Maps in a Web App
Indoor Mapping Data Format (IMDF)

Indoor Maps Program

Displaying Indoor Maps in an iOS App

Displaying Indoor Maps in a Web App
Indoor Map
Indoor Map

Building Footprint
Indoor Map

Doors
Indoor Map

Kiosks
Indoor Data Mapping Format (IMDF)

Data model for indoor spaces

2D format based on GeoJSON (RFC-7946)
fixture.geojson
detail.geojson
building.geojson
anchor.geojson
venue.geojson
unit.geojson
opening.geojson
level.geojson
footprint.geojson
address.geojson
manifest.json
{}
//snippet from level.geojson

"type": "Feature",

"geometry": {
  "type": "Polygon",
  "coordinates": [
    [[-123.4293921, 48.6406522], ...]
  ]
},

"properties": {
  "category": "unspecified",
  "outdoor": false,
  "ordinal": 0,
  "name": { "en": "Level 1" },
  "short_name": { "en": "L1" },
  ...
  "building_ids": [ "0c36de8f-dcee-4c48-b011-2c27c7f55933" ]
},

"id": "139c9194-36d7-424c-9fd0-5524c2d9406f",

"feature_type": "level"
"feature_type": "level",
"geometry": { "type": "Polygon", ... }
"properties": {
  "category": "parking",
  "ordinal": -1,
  "name": { "en": "Parking Level 1" },
  "short_name": { "en": "P1" },
  "building_ids": []
  ...
IMDF
Unit

"feature_type": "unit",
"geometry": { "type": "Polygon", ... }

"properties": {
  "category": "room",
  "level_id": "...",
  "name": { "en": "Storage Room 2" },
  "short_name": { "en": "SR2" }

  ...
}
"feature_type": "opening",
"geometry": {
  "type": "LineString",
  ...
}
"properties": {
  "category": "emergencyexit",
  "level_id": "...",
  "accessibility": "wheelchair",
  "access_control": "badgereader",
  ...
  "door": {
    "automatic": false
  }
}
"feature_type": "kiosk",
"geometry": {
    "type": "Polygon",
    ...
}
"properties": {
    "level_id": "...",
    "anchor_id": "..."
    ...
}
"feature_type": "occupant",
"geometry": null,
"properties": {
  "category": "localservices",
  "anchor_id": "...",
  "name": { "en": "Caffè Jurassic" },
  "hours": "Mo–Fr 10:00–21:30",
  "phone": "+14085512150",
  "website": "https://apple.com"
}
"feature_type": "anchor",
"geometry": {
    "type": "Point",
...
}
"properties": {
    "address_id": "...",
    "unit_id": "...
}
"feature_type": "amenity",
"geometry": {
  "type": "Point", ...
}
"properties": {
  "category": "exhibit",
  "unit_ids": ["..."],
  "name": { "en": "Exhibit Hall" },
  "hours": "Tu–Su 09:30–17:15",
  "accessibility": "wheelchair"
}
"feature_type": "section",
"geometry": {
    "type": "Polygon",
    ...
},
"properties": {
    "category": "postsecurity",
    "level_id": "...",
    "name": {
        "en": "Post Security 0"
    },
    "accessibility": "wheelchair"
}
"feature_type": "building",
"geometry": null,
"properties": {
  "category": "parking",
  "address_id": "...",
  "name": { "en": "Parking A" },
  "accessibility": "wheelchair"
}
IMDF
Footprint

"feature_type": "footprint",
"geometry": {
  "type": "Polygon", ...
}
"properties": {
  "category": "aerial",
  "building_ids": [...]
}
"feature_type": "footprint",
"geometry": {
    "type": "Polygon",
    ...
}
"properties": {
    "category": "ground",
    "building_ids": [...]
}
"feature_type": "footprint",
"geometry": {
    "type": "Polygon", ...
}
"properties": {
    "category": "subterranean",
    "building_ids": [...]  
}
Indoor Mapping Data Format

- Version: 1.0.0.beta.2
- Last update: March 04, 2019

Overview

Indoor Mapping Data Format (referenced throughout this document as "IMDF") provides a generalized, yet comprehensive model for any indoor location, providing a basis for orientation, navigation and discovery. In this release there are also detailed instructions for modeling the spaces of an airport and a shopping mall.

Developers can access both text and visual examples of all features, along with clear explanations of all terms. IMDF conforms to RFC 7946, ensuring compatibility and transferability of the data. IMDF is lightweight, mobile friendly, and can be rendered on any device, OS, or browser.

For GIS and BIM specialists, there is support for IMDF in many of your favorite tools.

IMDF is used by Apple to provide Indoor Positioning support with Core Location. Indoor maps integrated with indoor positioning can establish the foundation for a wide range of consumer and enterprise location-based apps and websites.
Indoor Mapping Data Format (IMDF)
Indoor Maps Program
Displaying Indoor Maps in an iOS App
Displaying Indoor Maps in a Web App
Indoor Maps

Easily create detailed maps of your indoor spaces and let visitors see where they are right in your app. Organizations with large public and private spaces like airports, shopping centers, arenas, hospitals, universities, and private office buildings can register for the Indoor Maps Program. Indoor maps are built using industry standard tools and require only your existing Wi-Fi network to enable GPS-level location accuracy so visitors can navigate your spaces with ease.

Create standards-based indoor maps

Indoor Mapping Data Format (IMDF) provides a generalized, yet comprehensive data model for any indoor location, creating a basis for orientation, navigation and discovery.

Enable indoor positioning

Apple’s indoor positioning technology passively uses your existing Wi-Fi infrastructure and requires no additional hardware deployment. This technology achieves GPS-level accuracy.
Join the Indoor Maps Program → Create IMDF → Validate IMDF → Enable Indoor Positioning → Integrate with Your App and Website → Building Owner
Integrate with your App and Website

Join the Indoor Maps Program

Create IMDF

Validate IMDF

Enable Indoor Positioning

Building Owner
Indoor Survey

Collect RF fingerprints
Indoor Survey

Collect RF fingerprints

Upload data for processing
Indoor Survey

Collect RF fingerprints
Upload data for processing
Test indoor positioning
Join the Indoor Maps Program

Create IMDF

Validate IMDF

Enable Indoor Positioning

Integrate with Your App and Website

Building Owner
Indoor Mapping Data Format (IMDF)
Indoor Maps Program
Displaying Indoor Maps in an iOS App
Displaying Indoor Maps in a Web App
Step 1: IMDF Model Classes

- venue.geojson
- building.geojson
- footprint.geojson
- level.geojson
- unit.geojson
- opening.geojson
- amenity.geojson
- anchor.geojson
- occupant.geojson

Venue | Building | Footprint
--- | --- | ---
Level | Unit | Opening
Amenity | Anchor | Occupant
Step 2: Decode IMDF

- venue.geojson
- building.geojson
- footprint.geojson
- level.geojson
- unit.geojson
- opening.geojson
- amenity.geojson
- anchor.geojson
- occupant.geojson

Diagram:

- MKGeoJSONDecoder
- Venue
- Building
- Footprint
- Level
- Unit
- Opening
- Amenity
- Anchor
- Occupant

Diagram shows the relationship between the IMDF components and the GeoJSON files.
Step 3: Render IMDF

Diagram showing the hierarchy of IMDF elements:
- Venue
  - Building
    - Footprint
    - Level
      - Opening
      - Unit
        - Amenity
        - Occupant

Map with various locations such as Restroom, Elevator, T-Rex, and other points marked.
// Step 1: IMDF Model Classes
// Feature.swift

class Feature<Properties: Decodable>: NSObject, IMDFDecodableFeature {
    let identifier: UUID
    let properties: Properties
    var geometry: [MKShape & MKGeoJSONObject]
}

class Unit: Feature<Unit.Properties> {
    ...
}

// Step 2: Decode IMDF
// Decoding IMDF file for Level feature

func decodeLevel(fileURL: URL) throws -> [Level] {
    let data = try Data(contentsOf: fileURL)
    let geoJSONDecoder = MKGeoJSONDecoder()
    let geoJSONFeatures = try geoJSONDecoder.decode(data)

    guard let features = geoJSONFeatures as? [MKGeoJSONFeature] else {
        throw IMDFError.invalidType
    }

    let levelFeatures = try features.map { try Level(feature: $0) }
    return levelFeatures
}
// Step 3: Render IMDF
// MapKit APIs for displaying overlays and annotations

// Add IMDF features as overlays and annotations to map
func addOverlays(_ overlays: [MKOverlay])
func addAnnotations(_ annotations: [MKAnnotation])

// MKMapViewDelegate protocol to customize appearance
func mapView(_ mapView: MKMapView, rendererFor overlay: MKOverlay) -> MKOverlayRenderer
func mapView(_ mapView: MKMapView, viewFor annotation: MKAnnotation) -> MKAnnotationView?
Demo
Displaying Indoor Map in an iOS Application

Create model classes

- Venue
- Building
- Footprint
- Level
- Unit
- Opening
- Amenity
- Anchor
- Occupant
Displaying Indoor Map in an iOS Application

Create model classes

Decode IMDF
Displaying Indoor Map in an iOS Application

Create model classes
Decode IMDF
Render IMDF
Displaying Indoor Map in an iOS Application

Create model classes
Decode IMDF
Render IMDF
Apply style
Displaying Indoor Map in an iOS Application

Create model classes
Decode IMDF
Render IMDF
Apply style
Display indoor location
// Step 4: Style

protocol StylableFeature {
    var geometry: [MKShape & MKGeoJSONObject] { get }
    func configure(overlayRenderer: MKOverlayPathRenderer)
    func configure(annotationView: MKAnnotationView)
}


Demo
Indoor Mapping Data Format (IMDF)
Indoor Maps Program
Displaying Indoor Maps in an iOS App
Displaying Indoor Maps in a Web App
// mapkit.importGeoJSON

let delegate = {
  geoJSONDidError(error, geoJSON) {},
  geoJSONDidComplete(items, geoJSON) {}
};

mapkit.importGeoJSON(geoJSON, delegate);
let delegate = {
  geoJSONDidComplete(items, geoJSON) {
    map.showItems(items); // or addItems() if no need to center the map on these items
  }
};

mapkit.importGeoJSON(geoJSON, delegate);
let delegate = {
    geoJSONDidColumnInfo(error, geoJSON) { ... },
    geoJSONDidComplete(items, geoJSON) { ... },
    styleForOverlay(overlay, geoJSON) {
        overlay.style.strokeColor = "white";
        overlay.style.lineDash = [3, 3];
        return overlay.style;
    }
};

mapkit.importGeoJSON(geoJSON, delegate);
// itemForPoint

let delegate = {
  geoJSONDidError(error, geoJSON) { ... },
  geoJSONDidComplete(items, geoJSON) { ... },
  styleForOverlay(overlay, geoJSON) { ... },
  itemForPoint(coordinate, geoJSON) {
    const options = {
      title: "Title of the annotation",
      displayPriority: 500,
      color: "green"
    };
    return new mapkit.MarkerAnnotation(coordinate, options);
  }
};

mapkit.importGeoJSON(geoJSON, delegate);
Demo
Displaying Indoor Map in a Web App

Organize features by type and filter by level
Displaying Indoor Map in a Web App

Organize features by type and filter by level

Add a level picker and display features on level
Displaying Indoor Map in a Web App

Organize features by type and filter by level

Add a level picker and display features on level

Style units and openings
Displaying Indoor Map in a Web App

Organize features by type and filter by level
Add a level picker and display features on level
Style units and openings
Customize annotations
Indoor Mapping Data Format (IMDF)
Indoor Maps Program
Displaying Indoor Maps in an iOS App
Displaying Indoor Maps in a Web App
Summary
Indoor Maps — Best practices

Apply styling to match the visual style of your app or brand
Apply styling to match the visual style of your app or brand

Select distinct styles based on category
Summary
Indoor Maps — Best practices

Apply styling to match the visual style of your app or brand

Select distinct styles based on category

Use easily recognizable icons
Summary
Indoor Maps — Best practices

Apply styling to match the visual style of your app or brand

Select distinct styles based on category

Use easily recognizable icons

Adjust map detail based on the zoom level
Summary
Indoor Maps — Best practices

- Apply styling to match the visual style of your app or brand
- Select distinct styles based on category
- Use easily recognizable icons
- Adjust map detail based on the zoom level
- Provide context
Summary
Indoor Maps — Best practices

- Apply styling to match the visual style of your app or brand
- Select distinct styles based on category
- Use easily recognizable icons
- Adjust map detail based on the zoom level
- Provide context
- Enable indoor user location
## More Information

developer.apple.com/wwdc19/241

<table>
<thead>
<tr>
<th>Session Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>MapKit, MapKit JS, and Indoor Location Lab</td>
<td>Friday, 3:00</td>
</tr>
<tr>
<td>What’s New in MapKit and MapKit JS</td>
<td>WWDC 2019</td>
</tr>
<tr>
<td>Introducing the Apple Maps Program</td>
<td>WWDC 2019</td>
</tr>
</tbody>
</table>