**version 7/17**

**OpenSidewalks data schema**

The OpenSidewalks data schema is an unofficial draft standard for mapping pedestrian network information. It consists of data primitives, data definitions, data interactions, and a concrete schema.

**Data primitives**

Data primitives are the basic units used to derive the more complex data definitions used by OpenSidewalks. They are exactly the same primitives used by OpenStreetMap.

**Data definitions**

Data definitions are inspired by and usually compatible with the OpenStreetMap project, with a primary goal of the OpenSidewalks schema to be fully compatible with OpenStreetMap as quickly as possible (pending new tags acceptable to the community).

**Data interactions**

Data interactions refers to ways in which map data should be composed or collected: the often-missing instructions on how to map several different elements of the pedestrian network so that the data can be used in concert.

**The Schema**

The schema defined by OpenSidewalks is an extension of the OpenMapTiles schema, itself a simplification of OpenStreetMap tagging schemas. It uses the same licensing terms: CC-BY.

**Data primitives**

Pedestrian networks are graphical structures: they define not only the shape and properties of pedestrian spaces, but how they connect to one another.

OpenStreetMap uses graph structures as its core elements of mapping and the OpenSidewalks standard uses identical conventions for describing its data: tags, nodes, and ways.

**tags**

Tags are key-value pairs used to annotate metadata about the other data primitives. Both keys and values are typically textual, such as "highway" and "footway" in the key=value pair of highway=footway. By convention, tags are referred to by their parent key=value pairs, so to discuss footways one might start a topic on highway=footway.

Keys are an enum: they are pre-specified and only predefined keys should be used unless the schema is being extended for local or research purposes. The vast majority of values are also enums: prescribed categories of textual values. Some values may be numeric, such as the width of a sidewalk.

Tags may follow nesting structures by conventions. For example, highway=footway defines a footway and footway=sidewalk designates that the type of footway is a sidewalk. The strategy of reusing values as keys to further specify the category of the primary tag is known as subtagging. Another nesting strategy is to use namespaces, where colons are used to add additional information to a tag category. For example, streets often have multiple forms of lane annotation: lanes=3, lanes:forward=1, lanes:backwards=2 would refer to a 3-lane street with two lanes in one direction and two in the other, with direction of the way indicating forward vs. backward.

Tags must have these properties:

* a key
* a value

**nodes**

Nodes are the most basic data primitive in the schema. They have geographical coordinates (latitude, longitude), a unique identifier, and optionally one or more tags.

~~Nodes must either be disconnected and tagged, as in the case for trees, or referred to by at least one way, in which case tagging is optional.~~

When nodes are part of a linear **way,** they will be referred to by that way. In this case, tagging of the nodes is optional. When nodes are disconnected from **ways**, the nodes must be tagged. See: “Data Interactions – Nodes”.

Nodes must have these properties:

* A node identifier (ID).
* A latitude coordinate in WGS84.
* A longitude coordinate in WGS84.

Nodes may also have these properties:

* Any number of tags

**ways**

Ways are the linear data structure for OpenSidewalks and are used to represent all paths. Ways represent their geographical structures (lines, polygons) through reference to an ordered set of nodes and tagging conventions.

Ways take on semantics for mapping through the addition of tags: all ways should have at least one primary highway tag.

Ways must have these properties:

* A way identifier (ID).
* An ordered set of node references (node IDs).
* A highway tag of an allowed subclass.

**Data definitions**

**highways**

This category derives from the OpenStreetMap "highway" convention and is the primary signifier of all traversible paths. highway types are referred to using the subclass layer. Pedestrians can use a wide range of highways, even if they are not the primary intended traffic, so several highway classes are defined for OpenSidewalks:

* Streets for vehicles: streets where the primary traffic is vehicles like cars or trucks.
* Pedestrian streets: streets where the primary traffic is intended to be pedestrians.
* Cycling paths: paths intended primarily for cyclists.
* Designated footways: foot paths and sidewalks.

The full set of tags that can be applied to designate highways is described in the subclass layer.

**highways as linear features**

highways are usually linear features, similar to LineStrings in a GIS context or ways in OpenStreetMap: their geometric or geographic representation refers to an ordered set of coordinates, so they have both a shape and a direction along which they are drawn: a start coordinate, an end coordinate, and intervening coordinates to connect them.

**Special case: highways as areas**

Representing pedestrian spaces with polygonal data is less common, but still useful data. Examples where this is currently widely appropriate are plazas and large pedestrian streets. There is no standard approach by which these data are turned into routable structures, however, so for mapping useful information we recommend "double tagging": map areas and some canonical linear highways through them.

**Data interactions**

**Nodes as features**

Nodes can represent coordinates along a linear **way** and may optionally have unique properties (tags) of their own (example??). They can also be disconnected features that are representative of something that is happening along or within a certain distance of a **way (**i.e. curbs, utility hole, fire hydrants, poles), in which case they are required to have unique tags.

**highways as routable (graph) features**

A linear highway can be thought of as its own small graph within the larger graphical representation of the routable network: its coordinates are nodes and the connections between them are edges.

~~When treated as a graphical representations, a linear highway can be thought of as its own small graph: the coordinates are nodes and connections between them are edges. nodes will be referred to frequently in the OpenSidewalks schema, referring to unique coordinates that may be part of a linear highway and have unique properties (tags) of their own.~~

To be meaningful for routing purposes, highways should be connected. When represented by a graph data structure (as in OpenStreetmap), this means that every highway should share at least one coordinate pair (node) with another highway, ideally with coordinate pairs receiving a Shared Streets unique identifier (node reference). In a routing context, highway data structures will be split whenever an internal coordinate is shared by another highway.

**Mapping pedestrian paths**

**connecting within pedestrian paths**

The pedestrian highways network must consist of connected components, which includes

sidewalks, “footway links”, and crossings. If sidewalks are not connected to the crossings associated with them, a “footway link” should be used to connect them. Crossings should be connected to each other via the network as well.

**connecting highways with different intended forms of traffic**

Pedestrians interface with highways that serve a variety of transportation options as their primary traffic: pedestrians, cars, trucks, trains, and other large vehicles. When pedestrian highways cross other highways on the same x, y, and z-level, as in they physically intersect with other highways, they should share a node at the point of intersection.

~~cross other highways on the same x, y, and z-level, as in they physically intersect one another, they should share a node.~~

**Schema Layers**

**transportation**

This is the only way/linestring-based layer currently used by AccessMap, and it is a direct extension of the OpenMapTiles schema.

**Fields**

**description**

*Unique to OpenSidewalks*

This is *not* from the original OpenStreetMap data, but is a free-form text field derived from spatial and other metadata, e.g. using street data one might have a description of "NE of Main St".

**name**

*From OSM*

The primary name: in general, the most prominent signposted name or the most common name in the local language(s).

Allowable geometries:

* way

**water**

*From OSM*

Specification of a water body

Allowable geometries:

* way

**footway**

*Unique to OpenSidewalks*

The original value for footway on OpenStreetMap. This allows differentiation between different classes of footway.

Allowable geometries:

* way

Possible values:

* crossing
* sidewalk

**crossing**

*Unique to OpenSidewalks*

Values for the crossing key in OpenStreetMap as applied to ways. This is not the original value, as there is fragmentation in tagging standards regarding marked crossings and they all roughly mean the same thing. Therefore, uncontrolled and zebra are all converted to marked.

Allowable geometries:

* way

Possible values:

* marked
* unmarked
* island
* traffic\_signals
* kerb\_raised
* kerb\_lowered

**kerb\_raised**

*Unique to OpenSidewalks*

This is an inferred quantity based on either network analysis or spatial proximity of a crossing to lowered curbs, and should be interpreted as indicating that using this path requires crossing over a raised curb interface. This quantity is useful for visualization - which paths required traversing a curb?

Allowable geometries:

* way

Possible values:

* 1
* 0

**incline**

*Unique to OpenSidewalks*

This is *not* the original OpenStreetMap tag for incline, which would indicate the maximum incline over a path, but is instead an estimated minimum incline based on DEM data over the length of the path. The original OpenStreetMap incline tag is fairly rare on footways to begin with. The value represents a "rise over run" estimate, i.e. a fraction of elevation gain/loss versus distance, and is directional: a negative value indicates downhill in the direction of the way whereas a positive value indicates uphill.

Allowable geometries:

* way

**surface**

*Unique to OpenSidewalks*

This is the original surface key in the OpenStreetMap data. It indicates the surface of the way being traversed, such as concrete vs. grass.

Allowable geometries:

* way

Possible values:

* asphalt
* concrete
* gravel
* grass
* paved
* paving\_stones
* unpaved

**length**

*Unique to OpenSidewalks*

This is the calculated length of the way, in meters, according to the Haversine formula (Great-Circle Distance). This calculation is typically left up to consumers of geometry data, as the geometry is, itself, furnished for geometrical analysis. This is likely how AccessMap should also handle these data, but for now length is precalculated.

Allowable geometries:

* way

**foot**

*Unique to OpenSidewalks*

Original value of the foot key if it is set to yes or no. Possible values:

* 1
* 0

**opening\_hours**

*Unique to OpenSidewalks*

Original value of the opening\_hours tag.

Allowable geometries:

* node

Possible values:

* see specific format

**elevator**

*Unique to OpenSidewalks*

Whether the path uses an elevator for vertical movement, e.g. building paths.

Allowable geometries:

* way

Possible values:

* 1
* 0

**width**

*Unique to OpenSidewalks*

Original value of the width tag if it has no units (unit conversion not yet supported) - implied unit is meters.

Allowable geometries:

* way

Possible values:

* Any non-negative real value expressed in meters

**layer**

The original layer tag on OpenStreetMap, this refers to the relative z-order of map elements. This is an integer that can be negative, so e.g. layer=0 is above layer=-1

Allowable geometries

* way
* node.

Possible values:

* any integer value.

**service**

The original value of the service tag on OpenStreetMap, this refers to service ways, which includes driveways, parking lots, and alleys.

Allowable geometries:

* way

Possible values:

* alley
* crossover
* driveway
* parking\_aisle
* siding
* spur
* yard

**amenity**

*From OpenStreetMap*

Features that serve a functional purpose.

Allowable geometries:

* node

Possible values:

* bench
* waste\_basket
* telephone

**power**

*From OpenStreetMap*

For marking and tagging facilities for the generation and distribution of electrical power.

Allowable geometries:

* node

Possible values:

* pole

**emergency**

*From OpenStreetMap*

This key describe various emergency services.

Allowable geometries:

* node

Possible values:

* fire\_hydrant

**building**

*From OpenStreetMap*

Allowable geometries:

* node

Possible values:

* levels (???)
* name

**buildings**

*From OpenStreetMap*

Allowable geometries:

* way

Possible values:

* yes (???)

**level (not used)**

*From OpenStreetMap*

This is not yet used, but should be in future specs. It is an integer indicating the associated building level of a feature, including footways - e.g. a pathway around part of a building.

Allowable geometries:

* node
* way

Possible values:

* Any integer value

**landuse**

*From OpenStreetMap*

For describing the primary use of areas of land.

Allowable geometries:

* node ??

Possible values:

* grass

**smoothness**

*From OpenStreetMap*

Provide a classification scheme regarding the physical usability of a way for wheeled vehicles.

Allowable geometries:

* way

Possible values:

* excellent
* good
* intermediate
* bad

**manhole**

*From OpenStreetMap*

Hole with a cover that allows access to an underground service location, just large enough for a human to climb through

Allowable geometries:

* node

Possible values:

* shape = circular/rectangular
* material=wood/metal/steel/concrete/stone/reinforced\_concrete

**public\_transport**

*From OpenStreetMap*

Public transport infrastructure tag

Allowable geometries:

* node

Possible values:

* stop\_position=right, left, turnout

**traffic\_signals**

*From OpenStreetMap*

Gives details about the type or function of traffic signals.

Allowable geometries:

* node

Possible values:

* sound=yes, no, unknown
* vibration=yes,no,unknown
* button\_operated=yes,no,unknown
* timing=???
* foot=yes,no,unknown

**brunnel**

This contains information on the bridge/tunnel/ford tags.

Allowable geometries:

* way

Possible values:

* bridge
* ford
* tunnel

**relation**

type=intersection

associatedstreet:count=

the relation will “collect” nodes and ways that belong to the intersection (including both sidewalks and crossings, islands, etc

Allowable geometries:

* node

Possible values:

* Intersection=any integer value

**junction**

Describing the kind of a junction.

Allowable geometries:

* node

Possible values:

* Roundabout

**indoor**

Whether the pathways is indoor or not (this is not yet implemented). Possible values are stated to be '1' in the OpenMapTiles spec - presumably the key doesn't exist at all if indoor=no.

Allowable geometries:

* way

Possible values:

* 1

**ramp (not used)**

This is an ambiguous / undefined tag for pedestrian ways and is *not* currently used by Accessmap. We do not use this OpenMapTiles field.

**subclass**

The highway=\* subclass of the displayed way. Many values are accepted, enumerated below by primary intended use case:

**cyclists:**

* cycleway: a cycling path. May have foot=yes or foot=no to explicitly designate pedestrian access.

**pedestrians:**

* footway: a pedestrian path. Includes sidewalks and detached footways.
* path: a catch-all for paths. Note: may be deprecated in favor of less ambiguous tags.
* pedestrian: a pedestrian street or area. A road where pedestrians are the primary traffic and other traffic is limited or prohibited.
* steps: stairs.

**motor vehicles:**

* secondary
* tertiary
* residential
* service

There are plans to add corridor for indoor routing.

**barriers**

*Unique to OpenSidewalks*

This is a layer not found in the OpenMapTiles spec that should, ideally, be contributed back into separate appropriate layers. It is focused on pedestrian features that pose potential barriers (though depending on the values, some are helpful infrastructure). Note that all features in the barriers layer are points.

Allowable geometries:

* node

Possible values:

* hedge
* wall
* fence
* kerb

**Fields**

**kerb**

The value of the original kerb tag.

Allowable geometries:

* node

Possible values:

* flush
* lowered
* raised
* rolled

**tactile\_paving**

The value of the original tactile\_paving tag.

Allowable geometries:

* node
* way

Possible values:

* 1
* 0

To discuss:

* what is the standard for tagging multiple values? i.e. crossing is marked and traffic signals. semicolons?
* need to further discuss “lighting” feature.
  + street\_lamp node
* nesting tags: subtagging or namespaces (using colons)? example: building and level, manhole and shape/material. how should these be organized in the schema?
* determine what key belongs to “maxspeed” value. highway=???
  + same with “steps” and “step\_count", “handrail”, “width”. barrier? highway?
* omitted ‘construction’ due to data quality issues
* handrail=both
* highway=steps
* incline=down