A geospatial API for linking and distributing open data - researching new ways of linking similar objects from different data sets

Abstract

The CitySDK Mobility API is a layer-based data distribution and service kit for geospatial Linked Open Data. The API distributes data from different organizations, silos and data sets in a unified way, and lets data owners annotate and add meaning to their data, as well as create links between different data sets. Linking data sets is important when multiple data sets share data about the same object. The API currently creates *hard links* by connecting two URIs, but this approach is not very flexible: when data can be added, removed or changed from any data sets, and URIs can change, too. This makes keeping links up to date cumbersome and prone to errors. We want to investigate new ways of linking similar objects from different data sets, by using machine learning and by using URIs that do not directly link to objects, but to *concepts*.

CitySDK

The CitySDK Mobility API, developed by Waag Society is a layer-based data distribution and service kit. Part of CitySDK, a European project in which eight cities (Manchester, Rome, Lamia, Amsterdam, Helsinki, Barcelona, Lisbon and Istanbul) and more than 20 organisations collaborate, the CitySDK Mobility API enables easy development and distribution of digital services across different cities. Originally, the CitySDK Mobility API was designed and developed to be used for *mobility* data (e.g. public transport, parking and road network data , but the API can be - and is - used for almost all data sets that share information about objects that exist in a city, e.g. have a geospatial component.

Like systems like CKAN, the CitySDK Mobility API has a data catalog function. But next to being a catalog, CitySDK does more. It acts as a proxy for data sets and stores data itself, and lets data owners annotate and add meaning to their data sets, using the Linked Open Data standards. The API exposes all its data sets in a unified way, and allows users of the API to access all data in both (Geo)JSON, and RDF/Turtle.

The API structures all data in the same way, this is reflected by a simple data model used in the API's database:

- All objects with a geographic location are called a `node`, with a unique URI;
- Per `layer`, `data` can be added to those nodes;

- This data is a key/value store, each layer can have its own keys and values.

For example:

- A museum *physically* exist in a city, and can be given a unique URI;
- Many organizations have some data about this museum;
- This museum exists as a `node` in the API;
- All organizations that have data about this museum, add this data on their own `layer`.

A graphical explanation of the CitySDK Mobility API concept can be found on the developer's site.

Linking data sets

Linking data sets is important when multiple data sets share data about the same object. While developing the CitySDK Mobility, we populated the system with any different Dutch data sets, and we also tried to automate the process of creating links between them. The following examples illustrate some of the data sets (and links between them), currently available via the API:

- OpenStreetMap is a valuable source of road network and POI data. We expose parts of the OSM database via CitySDK, and we link real-time events from the Dutch bureau for tourism to OSM POIs.
- The Dutch cadastre keeps an open data set of all adresses and buildings in the Netherlands. Many other data sets in the Netherlands have data available per address, for example the database of Dutch national heritage sites. In the API, we link both data sets by using the address URI.
- A country-wide data set is available for all neighbourhoods in the Netherlands, each with a unique ID. To those neighbourhoods, we link statistical data sets as well as real-time weather predictions.

The API currently links similar objects from different data sets by creating *hard links* between those objects. Each node has a unique URI, and data from different data sets can be linked to this URI, each on its own layer. This approach has some drawbacks, and the following questions can be asked:

- What happens to links when data from one ot the data sets changes?
- How do you automate the creation of links? And who is responsible?
- When exactly are two objects *the same*?

Further research - machine learing and *concepts*

While working on a new version of the CitySDK Mobility API, we want to research new ways of linking similar objects from different data sets. Automating the creation of links between those objects need to be investigated further:

1. Currently, objects are linked on *insertion time*; when they are added to the API's database. Can we design a system which only creates links at *query time*?
2. Objects (and data from linked data sets) can be accessed by the objects' URI. Would it be possible to design a new URI scheme, which does not point to individual objects, but to *concepts*? Instead of using a URI for a bus stop from one data set, we could think of a URI which points to a more fuzzy concept of a bus stop, around a certain location and with a certain name, all within certain thresholds.
3. Does this new API and URI scheme need its objects to be categorized? How do we start categorizing objects from a wide variaty of data sources, and what can we learn from previous attempts?
4. Which machine learning techniques can we use to automate the process of creating links?
5. Can we design easy-to-use UIs to help data owners importing data, as well as automatically see overlap between other existing data sources?

Links

- CitySDK Mobility API
- Explanation of concept and data model
- Map of all 9,866,539 buildings in the Netherlands
- Visualisation of real-time public transport data