LAND SCAN API

OVERVIEW

Create an API using Nodejs that allows the user to interact with the API and get appropriate details with proper token-based authentication and authorization.

SCENARIO

You are being asked to solve a problem related to **GIS** (Geographical Information Systems). Assume that we have a GIS dataset containing a georeferenced, high-resolution map of a region. This dataset also includes **vectors** that are drawn over the region. Both the map and the vectors have a certain set of attributes associated with them.

For example, consider Bangalore as a region and there can be certain polygons drawn on its map, like houses, parking areas, shopping malls, etc. So each of these polygons will have a class id and class name. These class id and class names are just a mapping to each other. For example, a house can have a (class id, class name) = (101, house).

So, each of these polygons will have certain additional attributes that are computed run time, i.e they should not be in the database. They are like **area, perimeter, center of mass,** etc.

SPECIFICATIONS

1. A region will have the following attributes.
    a. UID* (different from the primary key)
    b. Name*
    c. Description
    d. Location* (geojson of type Point)
    e. Owner* (a reference/foreign key to user object)
2. A vector/polygon will have the following attributes.
    a. UID* (different from primary key)
    b. Name*
    c. Description
    d. ClassID* (a numeric identifier of your choice)
    e. ClassName* (a name for the tag of a vector)
    f. Polygon* (geojson of type Polygon)
    g. Region* (a reference/foreign key to a region)

h. Owner* (a reference/foreign key to user object)

Additionally, a vector/polygon should have certain computed properties like area, perimeter which should not be a part of the database.

API SPECIFICATION

USERS API

1. User creation (a token should be created once an account is created.)
2. User login (as a response the associated token should be returned.)

REGIONS API

1. Basic CRUD operations.

VECTORS API

1. Basic CRUD operations.
2. An endpoint to filter vectors by (mutually inclusive) (as query parameters)
    a. Class Name
    b. Area
    c. Perimeter
    d. Region (UID)
    e. Polygon (a geojson to filter vectors which fall inside this polygon)

Note: All the CRUD operations require a user token as an authorization header. And the owner is the user who creates the resource. The UID should be created automatically i.e, the user should not provide the UID. Only the owner can update/delete a resource.

MANDATORY TECH STACK

- Node.js
- MongoDB (Mongoose as ODM)
- Express/Hapi.js or similar framework.

TIPS

- To know more about geojson and its specification visit http://geojson.io/ and http://geojson.org/
- Consider using http://turfjs.org/

SUBMISSIONS

- Please submit your solutions to anubhutiv@sensehawk.com and saideep@sensehawk.com
- Make sure you are submitting the final version.
- Feel free to contact us on the emails shared above for any queries.