oOh! Media Coding Exercise

This exercise is intended to allow you to demonstrate your skills in building an small Javascript-based web API. Your submission will be assessed on the following criteria:

1. Grasp and understanding of the Javascript language
2. Writing clearing, concise, maintainable code
3. Ability to verify the correctness of the solution

The requirements are split into **MUST**'s: things we expect your solution to include, **SHOULD**'s: things we'd like to see you include, and **COULD**'s: are nice-to-have's and you can choose to do any, all, or none of them. You are encouraged to treat this as a microcosm of a real project, so approach it as you would any other project. Javascript is required, but other technologies and libraries are at your discretion.

Please include a README with your submission which describes how to run the project, and explains the approach you took to producing the solution

Submit your solution by either sending a link to a Github or Bitbucket repository containing your code, or a link to a zipped copy of your code in Dropbox or similar.

Exercise description

oOh! Media is designing a shopping centre inventory management system which will help their product management team maintain records of where physical display panels are installed in shopping centres. You are creating an API to manage inventory and shopping centres, allowing persisting and modifying data, as well as (optionally) a interface for users to manage the inventory.

Domain

A **Shopping Centre** is a mall like MYER or Westfield, with many stores operating inside; placed at key locations within the Shopping Centre are Assets, which are oOh! Media units for displaying content. Shopping Centres must at a minimum have the attributes: Name, address, and have Assets associated with them.

An **Asset** is a physical screen which receives advertisement and other content

throughout the day. It has physical attributes such as it's dimensions, a location within the shopping centre, and a status indicating whether it is active to receive content or offline for maintenance. Assets must at a minimum have the attributes: Name, physical dimensions, associated Shopping Centre, location within the centre, and status.

Your solution

1. **MUST** have an API server written in Javascript
2. **MUST** have routes for Shopping Centres
3. **MUST** have routes for Assets
4. **MUST** persist data to a database
5. **SHOULD** be secured against anonymous access and track which user makes changes to the data
6. **SHOULD** allow marking Assets "inactive" for when they're receiving maintenance, and re-activating them later
7. **COULD** have a UI (but don't worry about UX)
8. **COULD** support searching for Assets by Name, Shopping Centre, or Status