**Node.js, Express, MongoDB, Mongoose Exercise**

A store wants to use a RESTful Web Service that can

* Create/Retrieve/Update/Delete Products. The store has multiple products.
* Create/Retrieve/Update/Delete Reviews for products. A product has many reviews.

Build this API using Node.js + Express, along with MongoDB (accessed via Mongoose) as the database.

**Models**

The MongoDB database for this application has a single collection for Products. Since data is denormalized in a MongoDB database, every Product document has the associated reviews

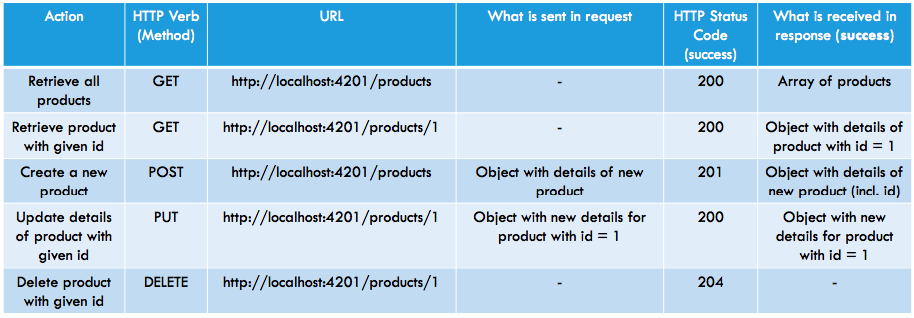
Product Model specifications

* \_id: ObjectID, Auto-generated
* name: String [required]
* description: String
* price: Number [required]
* rating: Number between 1 and 5
* urls: An array of image URLs (each URL is a String)
* reviews: An array where every item is a sub-document with the following structure
  + \_id: ObjectID, Auto-generated
  + reviewer: String [required]
  + rating: Integer between 1 and 5 [required]
  + title: String
  + text: String [required]
  + createdAt: Date, set automatically to the date review was created on [required]

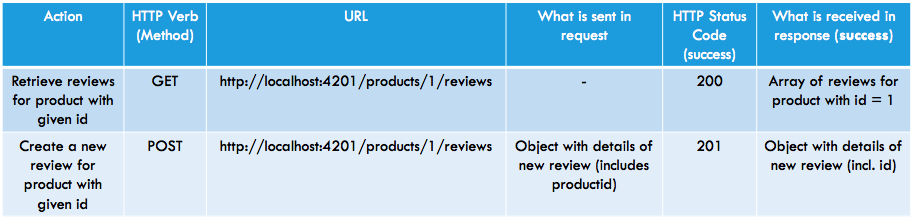
**Note**: You will need to define 2 schemas – one for Product, and one for Review and use the Review’s schema within Product’s. **Refer**: <https://mongoosejs.com/docs/subdocs.html>

**API**

The API has endpoints for working with products and reviews. The specifications are listed below.

Products Resource****

Reviews Resource



**Note**:

The tables show response codes in case of success only. The API will need to handle all error cases appropriately. Some of the popular codes in case of error are.

1. 400 – Bad Request. The data sent in the request was malformed.

2. 403 – Not Authorized

3. 404 – Resource not found

4. 409 – Duplicate resource exists and hence new resource was not created

5. 500 – Internal server error