Peer-graded Assignment: Assignment 2: MongoDB

Submit by November 25, 11:59 PM PST

Important Information
It is especially important to submit this assignment before the deadline, November 25, 11:59 PM PST, because it must be graded by others. If you submit late, there may not be enough classmates around to review your work. This makes it difficult - and in some cases, impossible - to produce a grade. Submit on time to avoid these risks.

Instructions

My submission

In this assignment, you will continue your journey with MongoDB and Mongoose. You will then create two new schemas for promotions and leadership, and then extend the Express REST API server to support the /promotions and the /leaders REST API end points.

Discussions

Step-By-Step Assignment Instructions less

Assignment Overview

At the end of this assignment you would have completed the following three tasks:

- Implemented the Promotions schema and model
- Implement a REST API to support the /promotions endpoint, and the /promotions/:promoId endpoint enabling the interaction with the MongoDB database
- Implemented the Leaders schema and model
- Implement a REST API to support the /leaders endpoint, and the /leaders/:leaderId endpoint enabling the interaction with the MongoDB database

Assignment Requirements

This assignment consists of the following two tasks:

Task 1

You are given the following example of a promotion document. You will now create the Promotions schema and model to support the document:

```
{
      "name": "Weekend Grand Buffet",
      "image": "images/buffet.png",
      "label": "New",
      "price": "19.99",
      "description": "Featuring . . .",
      "featured": false
}
```

Note in particular that the label and price fields should be implemented the same way as you did for the Dishes schema and model. The Promotions schema and model should be defined in a file named *promotions.js*.

Next, extend the *promoRouter.js* to enable the interaction with the MongoDB database to fetch, insert, update and delete information.

Task 2

You are given the following example of a leadership document. You will now create the Leaders schema and model to support the document:

```
{
      "name": "Peter Pan",
      "image": "images/alberto.png",
      "designation": "Chief Epicurious Officer",
      "abbr": "CEO",
      "description": "Our CEO, Peter, . . .",
      "featured": false
}
```

The Leaders schema and model should be defined in a file named *leaders.js*.

Next, extend the *leaderRouter.js* to enable the interaction with the MongoDB database to fetch, insert, update and delete information.

Review criteria less

Upon completion of the assignment, your submission will be reviewed based on the following criteria:

Task 1

- The Promotions schema and model correctly supports all the fields as per the example document given above
- The label field is set to an empty string by default
- The price schema is be supported with a new SchemaType called Currency.
- The REST API endpoints /promotions and /promotions/:promoId are implemented to interact with the MongoDB database

Task 2

- The Leaders schema and model correctly supports all the fields as per the example document given above.
- The REST API endpoints /leaders and /leaders/:leaderId are implemented to interact with the MongoDB database