

# MongoDB and Mongoose I Coursera

## Important Information

It is especially important to submit this assignment before the deadline, March 27, 11:59 PM PDT, 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

### Assignment Overview

In this assignment, you will continue your journey with MongoDB and Mongoose. You will update the Dishes schema and model to support a few new fields in the schema. You will then create two new schemas for promotions and leadership, and test them by implementing a simple server. At the end of this assignment you would have completed the following three tasks:

- Updated the Dishes schema and model to support some new fields and tested with a modified server.
- Implemented the Promotions schema and model, and tested with a new server
- Implemented the Leaders schema and model, and tested with a new server.

### Assignment Requirements

This assignment consists of the following three tasks:

#### Task 1

You are given the following example of a dish document. You will now updated the Dishes schema and model to support the new fields in the document:

```
{
  "name": "Uthapizza",
  "image": "images/uthapizza.png",
  "category": "mains",
  "label": "Hot",
  "price": "4.99",
  "description": "A unique . . .",
  "comments": [
    {
      "rating": 5,
      "comment": "Imagine all the eatables, living in conFusion!",
      "author": "John Lemon"
```

```

    },
    {
      "rating": 4,
      "comment": "Sends anyone to heaven, I wish I could get my mother-
in-law to eat it!",
      "author": "Paul McVites"
    }
  ]
}

```

Note in particular, the new fields: image, category, label and price. The label field should be set to an empty string by default (check the Mongoose schema documentation to see how you specify default values). The price schema should be supported with a new SchemaType called Currency. This is supported through a Mongoose module called [mongoose-currency](#). It is fairly simple to use, and the basic use case described in the documentation is sufficient to support our price schema. The Dish schema should be defined in a file named *dishes.js*. Test your schema and model with an implementation of a server just like we did in the previous exercise.

## Task 2

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 . . ."
}

```

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*. Test your schema and model with an implementation of a server just like we did in the previous exercise.

## Task 3

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, . . ."
}
```

The Leaders schema and model should be defined in a file named *leadership.js*. Test your schema and model with an implementation of a server just like we did in the previous exercise.

Review criterialess

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

### Task 1

- The updated Dishes schema and model is correctly updated to support the new fields: image, category, label and price.
- The label field is set to an empty string by default
- The price schema is be supported with a new SchemaType called Currency.

### Task 2

- 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.

### Task 3

- The Leaders schema and model correctly supports all the fields as per the example document given above.