

Programming Question

Complete the code for the following Express Application:

You have to build an application for car rental service. You will be responsible for building the backend API for the company.

Assume that your MongoDB collection has the following structure:

```
cars {  
  _id: ObjectId(),  
  brand: String,  
  type: String,  
  year: String,  
  status: Number,  
  rate_per_day: Number,  
  rental_details: [{driver_name: String,  
                    reservation_id: String,  
                    driving_license: String,  
                    start_mileage: Number,  
                    end_mileage: Number,  
                    number_of_days: Number,  
                    total_rent: Number}, ...]  
}
```

A car status indicates whether a car is available (1) or rented (0).

Example:

```
cars {  
  _id: ObjectId(09384572039845203894752038942570),  
  brand: "Toyota",  
  type: "Camry",  
  year: "2017",  
  rate_per_day: 45,  
  status: 1,  
  rental_details: [  
    {driver_name: 'Asaad Saad',  
      reservation_id: ObjectId(0938457203984520389387456398475639),  
      driving_license: 'AL2019B5',  
      start_mileage: 15000,  
      end_mileage: 15500,  
      number_of_days: 5,  
      total_rent: 225 }...]
```

- **Note: calling ObjectId() will return a new ID to be used.**
- **No error handling is required,**
- **No input validation is required.**
- **Syntax is flexible**
- **Logic is very important**

Requirements:

1. When users send a GET request to `/cars` you should do the following:

- Read all the available `cars` and send a JSON response with the car specification details (brand, type, year, status and rate per day).

2. When users submit their reservation details (name, driving license), you will receive their data through a POST request to `/cars/:id/reserve`

Perform the following:

- Get the last mileage on the car from the last rental record and use it as `start_mileage`.
- Insert a new reservation record with all (name, driving license, `start_mileage` and a new generated `reservation_id`).
- Upon success, send them their confirmation id as JSON
`{success:1, reservation_id: ObjectId(093884520389387456398475639)}`

3. When users send their total number of rental days and end mileage value in a PATCH request to `/cars/:id/reserve/:reservation_id`

Perform the following:

- Update the rental record with the received data along with the total rent.
- Upon success: send the user the following JSON response:
`{success:1, total_rent: 225}`

4. When users send a DELETE request to `/cars/:id/reserve/:reservation_id` you should do the following:

- Remove the rental record from the DB.

You are required to use Express Router and complete the code in both `app.js` and `/routes/cars.js` files.

There is no need to handle any input validation or add any security configuration in your application.

app.js

```
var express = require('express');
var path = require('path');
var cars = require('./routes/cars');

var app = express();

app.use(express.json());
app.use(express.urlencoded({ extended: false }));


app.use('/cars', cars);

app.use(function(err, req, res, next) {
  res.status(err.status || 500);
  res.render('error', {
    message: err.message,
    error: {}
  });
});
app.listen(3000);
```

`/routes/cars.js`

`/routes/cars.js`