



Jeff Song <jefftimes@gmail.com>

Re: [mongodb-developer/mern-stack-example] revise the code for findOne() by _id in server/record.js (#5)

Jeff Song <jefftimes@gmail.com>

Fri, Sep 17, 2021 at 1:30 PM

To: mongodb-developer/mern-stack-example

<reply+AMQEJG5O2CZFORMPVNHUXJV7KB44JEVBNHHDWDSJUM@reply.github.com>

Cc: mongodb-developer/mern-stack-example <mern-stack-example@noreply.github.com>, Mention <mention@noreply.github.com>

Hi Joel

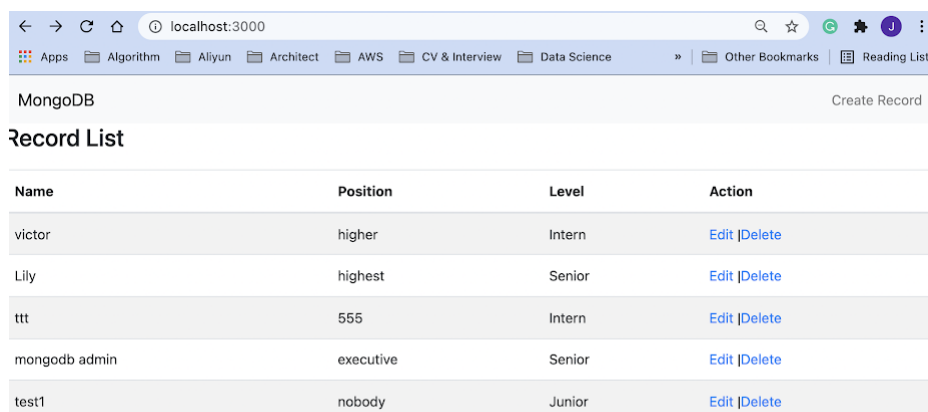
You will not receive any error but the logic of the code is incorrect, let me explain it to you. And I am not sure if it is a bug from the findOne() API from the mongodb nodejs package. the findOne() will always return the first document if we pass it a query object with the property set to undefined.

The original code snippet for the '/record/:id' path is showing below, it passes the `req.body.id` into 'myquery' object { id : `req.body.id` } as the value for the 'id' property and fire the query, then respond a result in JSON.

// This section will help you get a single record by id

```
recordRoutes.route("/record/:id").get(function (req, res) {  
  let db_connect = dbo.getDb();  
  let myquery = { id: req.body.id };  
  db_connect  
    .collection("records")  
    .findOne(myquery, function (err, result) {  
      if (err) throw err;  
      res.json(result);  
    });  
});
```

But when I created few records for the App and try to click the 'edit' anchor of any record, in the 'edit' page, I always get the first record showed, no matter what record I had chosen from.



Name	Position	Level	Action
victor	higher	Intern	Edit Delete
Lily	highest	Senior	Edit Delete
ttt	555	Intern	Edit Delete
mongodb admin	executive	Senior	Edit Delete
test1	nobody	Junior	Edit Delete

Base on my understanding of the 'named segments' pattern in ExpressJS, the id should be passed by the `req.params` object, but I am not hundred percent sure if there is some data in the req.body that could be used. So I add 4 logs for further understanding of what happened.

// This section will help you get a single record by id

```
recordRoutes.route("/record/:id").get(function (req, res) {

  let db_connect = dbo.getDb();
  let myquery = { id: req.body.id };

  console.log(' req.params.id in "/record/:id" path: ', req.params.id);
  console.log(' req.body.id in "/record/:id" path: ', req.body.id);
  console.log('myquery obj in "/record/:id" path: ', myquery);

  db_connect
    .collection("records")
    .findOne(myquery, function (err, result) {
      if (err) throw err;

      console.log('result in "/record/:id" path: ', result);

      res.json(result);
    });
});
```

The log confirm my suspicion that the req.body is empty, and the id is passed into by the req.params object. But something interest me is the findOne() returned the first record from the collection no matter it takes an object which id property is undefined. It behaves not as it should be.

```
[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server is running on port: 5001
Successfully connected to MongoDB.
  req.params.id in "/record/:id" path: 61384e33dac32e3ab833bf19
  req.body.id in "/record/:id" path: undefined
myquery obj in "/record/:id" path: { id: undefined }
result in "/record/:id" path: {
  _id: 6137f0137015894a7cd4d9a0,
  person_name: 'victor',
  person_position: 'higher',
  person_level: 'Intern'
}
```

I take the further action to pass the right data from the req.params.id to myquery's id property, and try it again. That is not what I expected, the frontend is yelling me an 500 error, log shows that findOne() return me a null.

```
✖ Failed to load resource: the server responded with a status of 500 (Internal Server Error) :5001/record/61384e3...:1
✖ Failed to load resource: the server responded with a status of 500 (Internal Server Error) :5001/record/61384e3...:1
```

```

[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server is running on port: 5001
Successfully connected to MongoDB.
  req.params.id in "/record/:id" path: 61384e33dac32e3ab833bf19
  req.body.id in "/record/:id" path: undefined
myquery obj in "/record/:id" path: { id: '61384e33dac32e3ab833bf19' }
result in "/record/:id" path: null

```

I turned into the database to have a looking what's the problem the findOne() return a null with a well set query. Then I found there isn't any property named as "id", only "_id" property there.

```

{
  "_id": {
    "$oid": "6137f0137015894a7cd4d9a0"
  },
  "person_name": "victor",
  "person_position": "higher",
  "person_level": "Intern"
}

```

But I stil got a null result even I changed the 'id' property of myquery object to '_id', because the value of the "_id" is not a string, it is an object - generated by ObjectId() API.

I finally got the edit initial page right after I revised the definition of myquery as below

/ This section will help you get a single record by id

```
recordRoutes.route("/record/:id").get(function (req, res) {
```

```
  let db_connect = dbo.getDb();
```

```
  // take _id as the property name, and assign a value returned by ObjectId();
```

```
  let myquery = { _id: ObjectId(req.params.id) };
```

```
  console.log(' req.params.id in "/record/:id" path: ', req.params.id);
```

```
  console.log(' req.body.id in "/record/:id" path: ', req.body.id);
```

```
  console.log('myquery obj in "/record/:id" path: ', myquery);
```

```
  db_connect
```

```
  .collection("records")
```

```
  .findOne(myquery, function (err, result) {
```

```
    if (err) throw err;
```

```
    console.log('result in "/record/:id" path: ', result);
```

```
    res.json(result);
```

```
  });
```

```
});
```

```

[nodemon] restarting due to changes...
[nodemon] starting `node server.js`
Server is running on port: 5001
Successfully connected to MongoDB.
req.params.id in "/record/:id" path: 6144c3120918a11ba0a49393
req.body.id in "/record/:id" path: undefined
myquery obj in "/record/:id" path: { _id: 6144c3120918a11ba0a49393 }
result in "/record/:id" path: {
  _id: 6144c3120918a11ba0a49393,
  person_name: 'mongodb admin',
  person_position: 'executive',
  person_level: 'Senior'
}

```

the same issue occurred in the '/update/:id' and '/delete/:id' path. so I correct all of them as showed below to make the App work properly. Otherwise you will always update or delete the first record.

// This section will help you update a record by id.

```

recordRoutes.route("/update/:id").post(function (req, res) {
  let db_connect = dbo.getDb("employees");
  let myquery = { _id: ObjectId(req.params.id) };
  let newvalues = {
    $set: {
      person_name: req.body.person_name,
      person_position: req.body.person_position,
      person_level: req.body.person_level,
    },
  };
  db_connect
    .collection("records")
    .updateOne(myquery, newvalues, function (err, res) {
      if (err) throw err;
      console.log("1 document updated");
    });
});

```

// This section will help you delete a record

```

recordRoutes.route("/:id").delete((req, res) => {
  let db_connect = dbo.getDb("employees");
  var myquery = { _id: ObjectId(req.params.id) };
  db_connect.collection("records").deleteOne(myquery, function (err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
  });
});

```

So, that's all. Hope it is helpful.

Jeff

[Quoted text hidden]