

Assignment on Express - Sequelize

(5) Define the table named Insurance and have the fields (a) PolicyNumber (Primary Key)

(b) Policy Holders Name (c) Policy Amt (d) Maturity Amount (e) Nominee

Define the following functions in express and data has to be fetched from DB using Sequelize.

(a) get - getAllPolicies - Retrieves all the policies in the table.

(b) get - getPolicyById - Retrieves the record based on the given policy id.

(c) post - newRecord - Inserts the data in the Insurance table.

(d) put - updatePolicy - Updates the policy record based on policy number.

(f) delete - deletePolicy - Deletes the policy record based on policyNumber.

(g) post - login - Checks whether the given user and password is valid or not by fetching records from USER table. User table will have UserID, Password and Role as the fields. Role will be either User or Admin.

(h) post - Register - Given data should be inserted into the USER Table.

(I) All the functions should be tested from postman.

(II) After user registers, check with login with the same data as given in register.

(III) Every function should return the status code along with the data. Status code should be verified in postman.

Status code 200 for if everything is fine.

Status code 400 if things are not fine.

Status code 401 if unauthenticated if login fails.

Status code 201 for the POST / PUT for insert and update transactions are performed successfully.

If you want to send the status from the server, you can use it this way

`res.status(200).send (data);` --> This will send the status 200 (Everything is fine)

(IV) If possible, Create a folder named Model and Place two js files in this. (a) Policy.js (b) Credentials.js.

Policy.js will have all the db related functions for policy. Credentials.js will have login and registration functions.

Export these modules and import them in the Express code and use it. All the routing code should be in Express code which makes use of these modules accordingly.

```
postgreAssignment / ... ass.js / ...
47 app.get('/getAllPolicy', (req, res) => {
48   InsuranceTable.findAll({raw: true}).then(data => {
49     console.log(data);
50     res.status(200).send(data);
51   }).catch(err => {
52     console.error("error is " + err);
53     res.status(400).send(err);
54   })
55 });
56
57
58 app.get("/getInsuranceById/:id", (req, res) => {
59   var id = req.params.id;
60   console.log("given id is " + id);
61
62   InsuranceTable.findByPk(id, {raw: true}).then(data => {
63     console.log(data);
64     res.status(200).send(data);
65   }).catch(err => {
66     console.error("error is " + err);
67     res.status(400).send(err);
68   });
69 });
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - postgreAssignment + -

Executing (default): SELECT "p_num", "P_H_Name", "Amount", "Maturity_amount", "Nominee" FROM "Insurances" AS "Insurances" WHERE "Insurances"."p_num" =

```
{
  p_num: 104,
  P_H_Name: 'omkar',
  Amount: 50000,
  Maturity_amount: 5000,
  Nominee: 'suraj'
}
```

Ln 47, Col 23 (12 selected) Spaces: 4 UTF-8 LF () Babel JavaScript Port: 55

Salesforce Training
Today at 4:00 PM

```
71 }));
72
73
74 app.use(express.json());
75 app.post("/InsertNewRecord", (req, res) => {
76     var p_num = req.body.p_num;
77     var P_H_Name = req.body.P_H_Name;
78     var Amount = req.body.Amount;
79     var Maturity_amount = req.body.Maturity_amount;
80     var Nominee = req.body.Nominee;
81
82     var insObj = InsuranceTable.build({p_num:p_num, P_H_Name:P_H_Name, Amount:Amount,
83     Maturity_amount:Maturity_amount, Nominee:Nominee});
84     insObj.save().then(data => {
85         console.log(data);
86         var strMsg = "record has been added into the table...";
87         res.status(201).send(strMsg);
88     }).catch(err => {
89         console.error("error is :"+err);
90         res.status(400).send(err);
91     });
92 });
```

node - postgresAssignment

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Nominee: 'vaibhav'
,
_previousDataValues: {
 p_num: 100,
 P_H_Name: 'navnath',
 Amount: 30000,
 Maturity_amount: 4000,
 Nominee: 'vaibhav'
},
,

Ln 108, Col 24 Spaces: 4 UTE-8 LF (.) Babel JavaScript Port: 55

</>

Send

Cookies

Beautify

200 OK 54 ms 323 B Save Response



15 JULY 2004

[Create Collection](#)

```

148 app.post("/login", (req, res) => {
149     var uid = req.body.uid;
150     var pass = req.body.pass;
151
152     fl = false;
153     const Op = Sequelize.Op;
154
155     usersTable.findAll({where: {[Op.and]: [{uid: uid}, {pass: pass}]},
156         raw: true,
157     }).then((data) => {
158         console.log(data);
159         console.log(typeof data);
160         if(data){
161             fl = true;
162         }
163     });
164     if(fl){
165         var strMsg = "your valid user";
166         res.status(201).send(strMsg);
167     } else {
168         var strMsg = "valid users enter correct details";
169         res.status(401).send(strMsg);

```

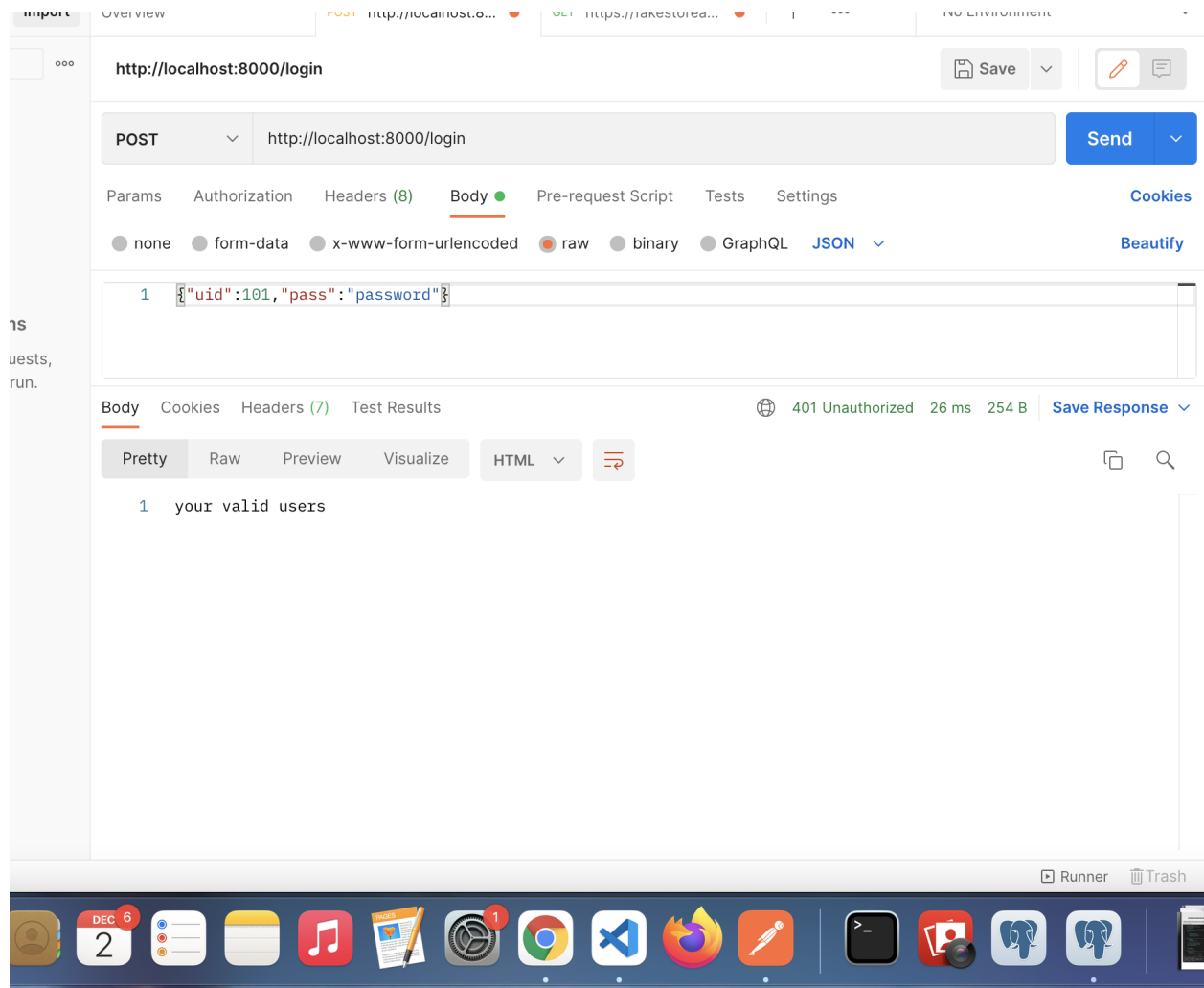
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

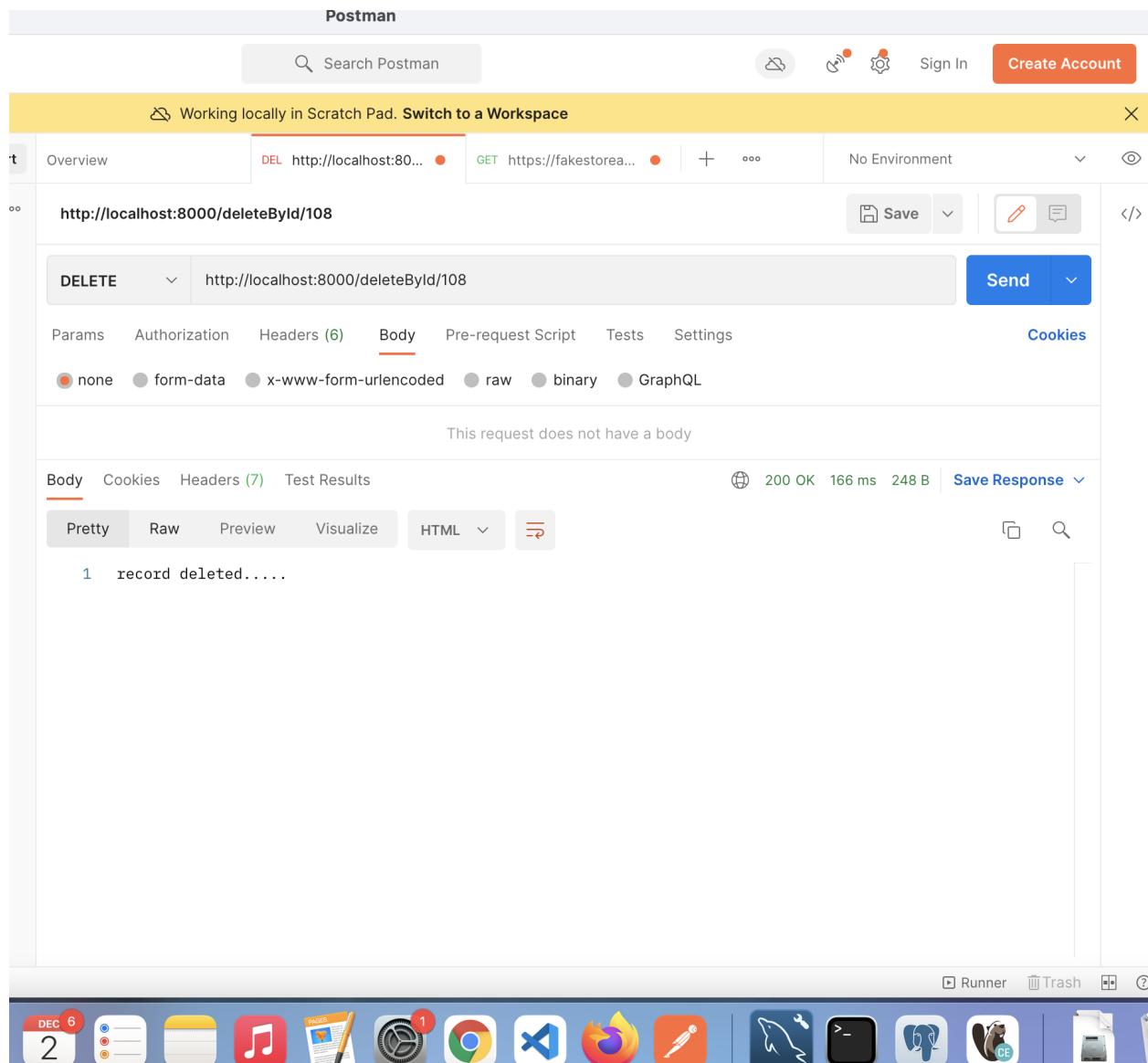
node - pos

```

server is listening at 8000
Executing (default): SELECT "uid", "pass" FROM "users" AS "users" WHERE ("users"."uid" = 101 AND "users"."pass" = 'password');
[ { uid: 101, pass: 'password' } ]
object
[nodemon] restarting due to changes...
[nodemon] starting `node ass.js`
server is listening at 8000

```





```
1 SELECT * FROM public."Insurances"  
2 ORDER BY p_num ASC
```

Data Output Explain Messages Notifications

	p_num [PK] integer	P_H_Name character varying (255)	Amount integer	Maturity_amount integer	Nominee character varying (255)
1	100	navnath	10000	8000	pranav
2	101	varun	20000	1000	navnath
3	102	sanket	30000	7000	varun
4	103	suraj	40000	8000	sanket
5	104	omkar	50000	5000	suraj
6	107	navnath	30000	4000	vaibhav

more

Search Postman



Sign In

Create Account

Working locally in Scratch Pad. Switch to a Workspace

Import

Overview

POST http://localhost:8...

GET https://fakestorea...

+ ...

No Environment

http://localhost:8000/login

Save



POST

http://localhost:8000/login

Send

Params

Authorization

Headers (8)

Body

Pre-request Script

Tests

Settings

Cookie

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 { "uid": 1011212, "pass": "password" }
```

ions

requests,
and run.

Body

Cookies

Headers (7)

Test Results



401 Unauthorized

5 ms

271 B

Save Response

Pretty

Raw

Preview

Visualize

HTML



```
1 valid users enter correct details
```

Runner

Trash



```

166         res.status(201).send(strMsg);
167     }else{
168         var strMsg ="valid users enter correct details";
169         res.status(401).send(strMsg);
170     }
171
172 })
173
174 app.post("/register", (req,res)=>{
175     var uid = req.body.uid;
176     var pass = req.body.pass;
177
178     var userObj = usersTable.build({uid:uid,pass:pass});
179     userObj.save().then(data=>{
180         var stri = "record is inserted";
181         res.status(201).send(stri);
182     }).catch(err=>{
183         console.error("error is :"+err);
184         res.status(400).send(err);
185     })
186 })
187

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

node - p

```

[nodeemon] restarting due to changes...
[nodeemon] starting `node ass.js`
server is listening at 8000
[nodeemon] restarting due to changes...
[nodeemon] starting `node ass.js`
server is listening at 8000
Executing (default): INSERT INTO "users" ("uid","pass") VALUES ($1,$2) RETURNING "uid","pass";

```

Ln 174, Col 20 (8 selected) Spaces: 4 UTF-8 LF ()

