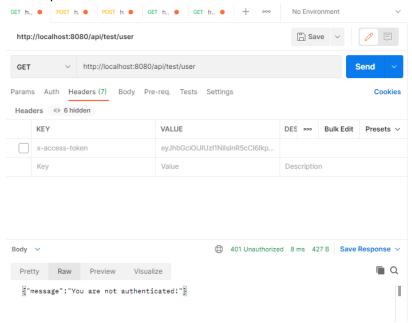
Name: Gordon Foo A0199554L

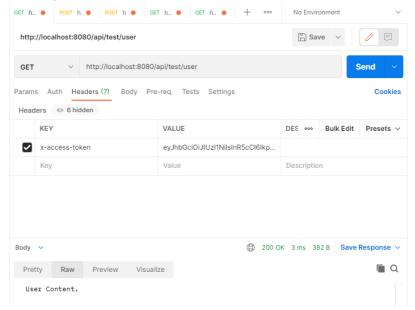
Link to github repo: https://github.com/gordonfgz/CS3219_OTOT_C.git

All credits go to: https://www.bezkoder.com/node-js-jwt-authentication-postgresql/

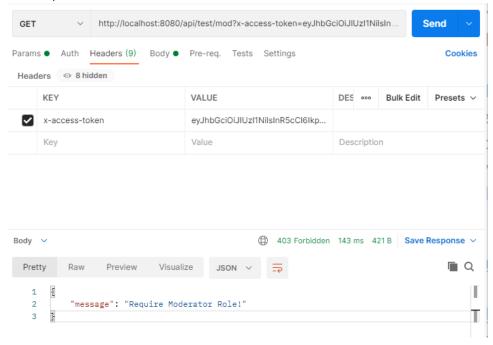
401 response on un authenticated user:



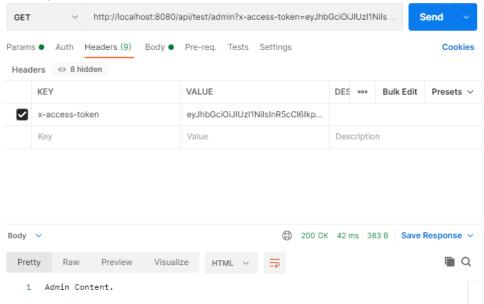
200 response on authenticated user:



403 response on authenticated but NOT authorised user:



200 response on authenticated and authorised user:



Scalability:

More roles can be added in the initialise method:

```
function initial() {
  Role.create({
    id: 1,
    name: "user"
  });

  Role.create({
    id: 2,
    name: "moderator"
  });

  Role.create({
    id: 3,
    name: "admin"
  });
}
```

Method to check can be implemented in authJWT

```
JS authJwt.js A X JS index.js ...\middleware A
middleware > JS authJwt.js > 쉾 verifyToken > 쉾 jwt.verify() callb
        res.status(403).send({
          message: "Require Admin Role!"
        });
        return;
      });
  isModerator = (req, res, next) => {
   User.findByPk(req.userId).then(user => {
      user.getRoles().then(roles => {
        for (let i = 0; i < roles.length; i++) {</pre>
          if (roles[i].name === "moderator") {
            next();
            return;
           }
        res.status(403).send({
          message: "Require Moderator Role!"
        });
    });
```