

Session maintenance using express and passport

```
var express = require("express");
var session = require('express-session');
var path = require("path");
var bodyParser = require("body-parser");
var mongoose = require("mongoose");
var app = express();
var router = express.Router();
var passport = require('passport');
var LocalStrategy = require('passport-local').Strategy;

app.use(session({
  secret: 'keyboard cat',
  resave: false,
  saveUninitialized: false
}));

var config = require("../config/config");
mongoose.connect(config.local.db, { useNewUrlParser: true });

require("../models/user");
var User = mongoose.model("User");

app.use(express.static(path.join(__dirname, "../dist/myapp/")));
app.use(bodyParser.urlencoded({ extended: false }));
app.use(bodyParser.json());
app.use(passport.initialize());
app.use(passport.session());

router.post("/api/registerUser", function (req, res) {
  var theUser = new User(req.body);
  theUser.save(function (err) {

    if (err) {
      console.log(err);
    } else {
      res.send(theUser);
    }
  });
});

router.get("/api/listusers", isUserAllowed(), function (req, res) {
```



```

//console.log(req.isAuthenticated());
//console.log(req.session);

User.find({}).exec(function (err, users) {
  if (err) {
    res.failure(err)
  } else {
    res.json(users);
  }
})
});

function isUserAllowed() {
  return function (req, res, next) {
    if (!req.isAuthenticated()) {
      res.redirect('/api/logout');
    }
    else {
      next();
    }
  }
}

router.get("/api/authStatus", function (req, res) {
  if (req.isAuthenticated()) {
    return res.status(200).json({
      status: true
    });
  } else {
    return res.status(500).json({
      status: false
    });
  }
});

router.get("/api/logout", function (req, res) {
  req.logout();
  res.json({ "message": 'success, you are now logged out!' });
});

router.get("/api/editUser/:id", isUserAllowed(), function (req, res) {
  User.findOne({ _id: req.params.id }).exec(function (err, user) {
    if (err) {

```



```

        res.failure(err)
    } else {
        res.json(user);
    }
})
});

router.put("/api/updateUser", isUserAllowed(), function (req, res) {
    var sid = req.body._id;
    User.findById(sid, function (err, doc) {
        doc.fname = req.body.fname;
        doc.lname = req.body.lname;
        doc.save(function () {
            res.json(doc);
        })
    })
});

router.delete("/api/deleteUser/:id", isUserAllowed(), function (req, res) {
    var sid = req.params.id;
    User.findById(sid, function (err, doc) {
        doc.remove(function () {
            res.json({ message: "deleted" });
        })
    })
});

router.param('userId', function (req, res, next, id) {
    User
        .findOne({
            _id: id
        })
        .exec(function (err, theUser) {
            if (err) return next(err);
            if (!theUser) return res.failure('Failed to load User ' + id);
            req.theUser = theUser;
            console.log("theUser in userId", theUser);
            next();
        });
});

router.post("/api/loginUser", function (req, res, next) {
    passport.authenticate('local', function (err, user, info) {
        if (err) {
            return next(err);
        }
    })
});

```



```

    }
    if (!user) {
      console.log(info.message);
    }
    req.logIn(user, function (err) {
      if (err) {
        return next(err);
      }
      req.theUser = user;
      res.send(user);

    });
  })(req, res, next);
});

//Use local strategy
passport.use(new LocalStrategy({
  usernameField: 'username',
  passwordField: 'password'
}),
function (username, password, done) {
  User.findOne({
    username: username
  }, function (err, user) {
    if (err) {
      return done(err);
    }
    if (!user) {
      return done(null, false, {
        message: 'Unknown user'
      });
    }

    if (!user.authenticate(password)) {
      return done(null, false, {
        message: 'Invalid passwordd'
      });
    }
    return done(null, user);
  });
});
});

//Serialize sessions
passport.serializeUser(function (user, done) {

```



```
//console.log('serialize user???');  
done(null, user.id);  
});  
  
passport.deserializeUser(function (id, done) {  
  User.findOne({  
    _id: id  
  }, function (err, user) {  
    if (user._id) user.userId = user._id;  
    done(err, user);  
  });  
});  
  
app.use(router);  
app.listen(4200);  
console.log("server is listening on port 4200");
```