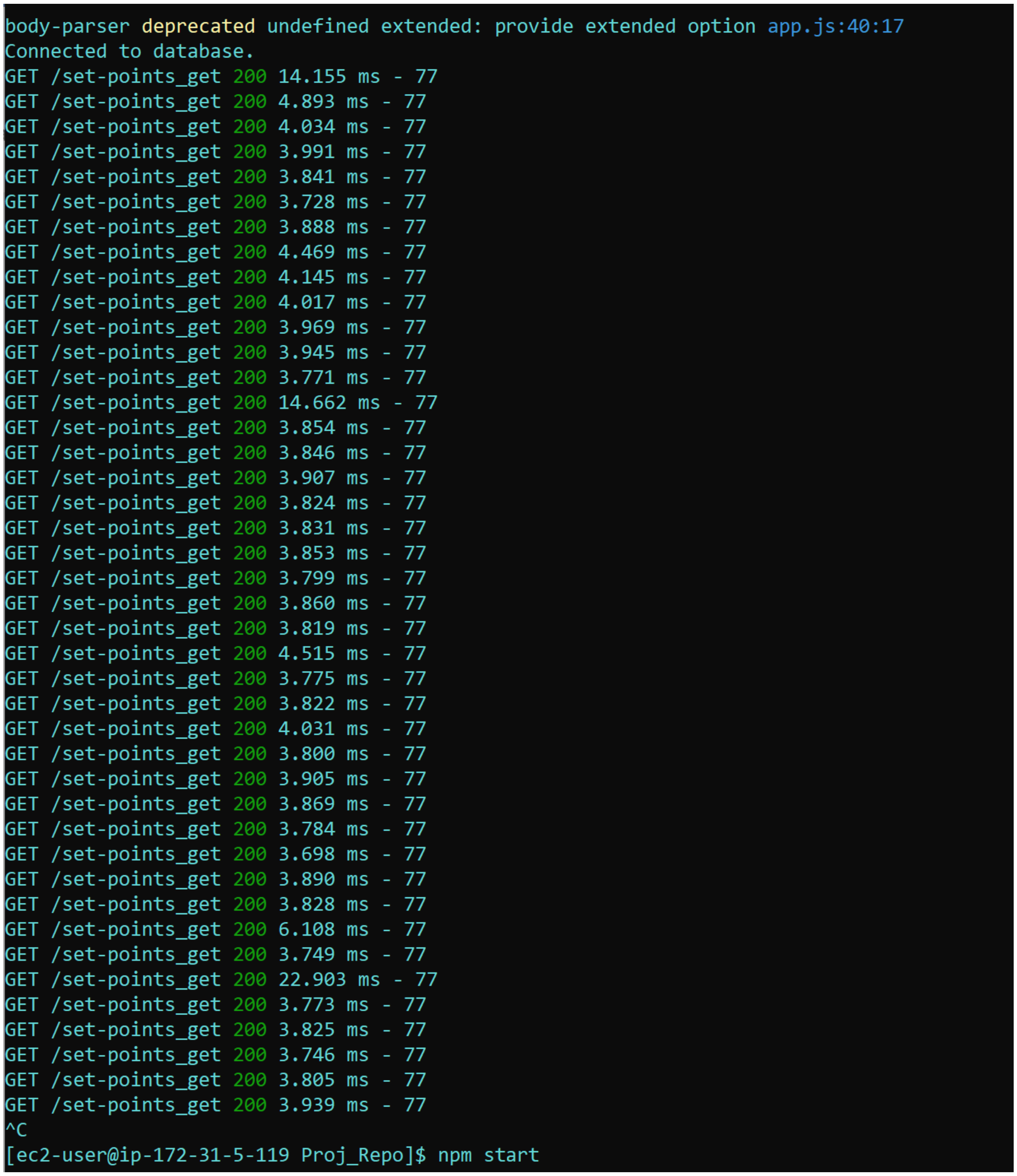
ECE 531: Introduction to the Internet of Things

Kevin Huang

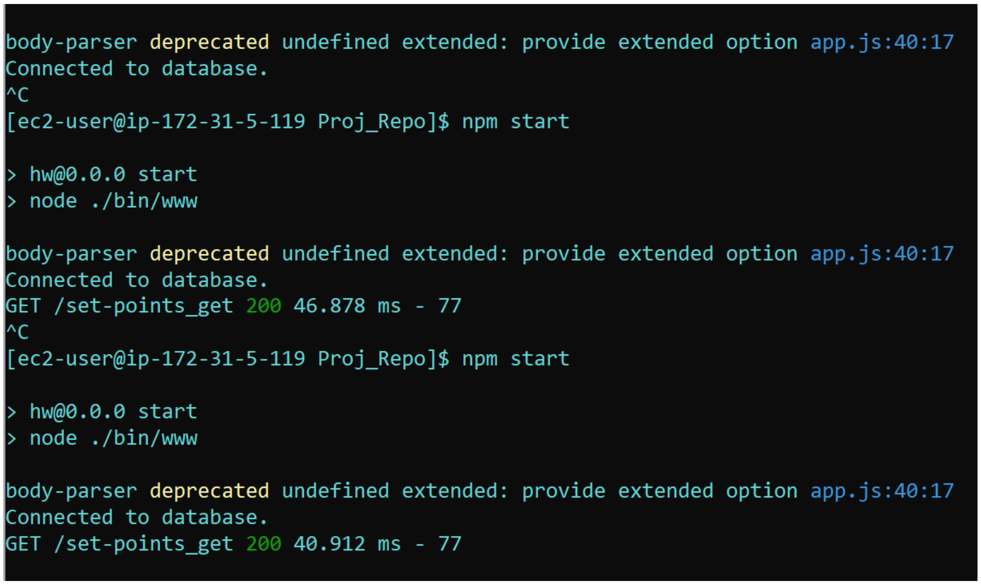
Final Project

My C program reads the thermocouple program, stores the temperature, then posts that temperature to the server where the function it accesses is routerpost(“/). From there, the current temperature was being compared to the high and low temperatures (temp1 & temp2) and I wanted the web server to make the decision to turn on or off the heater, and then send that status back to the C program. The C program does connect to the web server, but I encountered problems with the POST function.



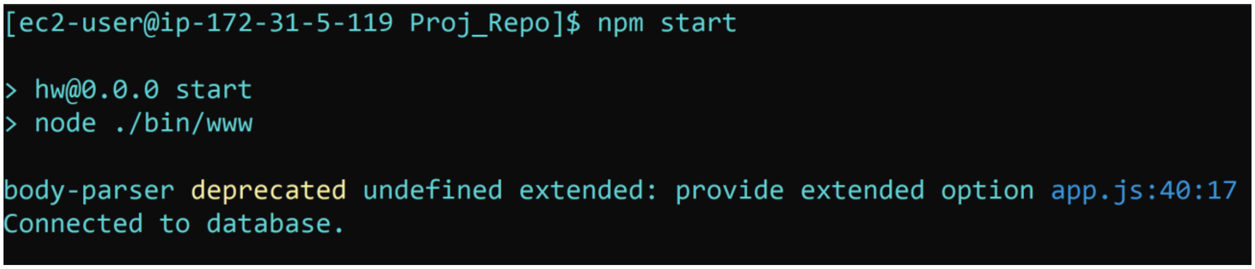
```
body-parser deprecated undefined extended: provide extended option app.js:40:17
Connected to database.
GET /set-points_get 200 14.155 ms - 77
GET /set-points_get 200 4.893 ms - 77
GET /set-points_get 200 4.034 ms - 77
GET /set-points_get 200 3.991 ms - 77
GET /set-points_get 200 3.841 ms - 77
GET /set-points_get 200 3.728 ms - 77
GET /set-points_get 200 3.888 ms - 77
GET /set-points_get 200 4.469 ms - 77
GET /set-points_get 200 4.145 ms - 77
GET /set-points_get 200 4.017 ms - 77
GET /set-points_get 200 3.969 ms - 77
GET /set-points_get 200 3.945 ms - 77
GET /set-points_get 200 3.771 ms - 77
GET /set-points_get 200 14.662 ms - 77
GET /set-points_get 200 3.854 ms - 77
GET /set-points_get 200 3.846 ms - 77
GET /set-points_get 200 3.907 ms - 77
GET /set-points_get 200 3.824 ms - 77
GET /set-points_get 200 3.831 ms - 77
GET /set-points_get 200 3.853 ms - 77
GET /set-points_get 200 3.799 ms - 77
GET /set-points_get 200 3.860 ms - 77
GET /set-points_get 200 3.819 ms - 77
GET /set-points_get 200 4.515 ms - 77
GET /set-points_get 200 3.775 ms - 77
GET /set-points_get 200 3.822 ms - 77
GET /set-points_get 200 4.031 ms - 77
GET /set-points_get 200 3.800 ms - 77
GET /set-points_get 200 3.905 ms - 77
GET /set-points_get 200 3.869 ms - 77
GET /set-points_get 200 3.784 ms - 77
GET /set-points_get 200 3.698 ms - 77
GET /set-points_get 200 3.890 ms - 77
GET /set-points_get 200 3.828 ms - 77
GET /set-points_get 200 6.108 ms - 77
GET /set-points_get 200 3.749 ms - 77
GET /set-points_get 200 22.903 ms - 77
GET /set-points_get 200 3.773 ms - 77
GET /set-points_get 200 3.825 ms - 77
GET /set-points_get 200 3.746 ms - 77
GET /set-points_get 200 3.805 ms - 77
GET /set-points_get 200 3.939 ms - 77
^C
[ec2-user@ip-172-31-5-119 Proj_Repo]$ npm start
```

Before I implemented the POST function, the C program and web server were communicating.

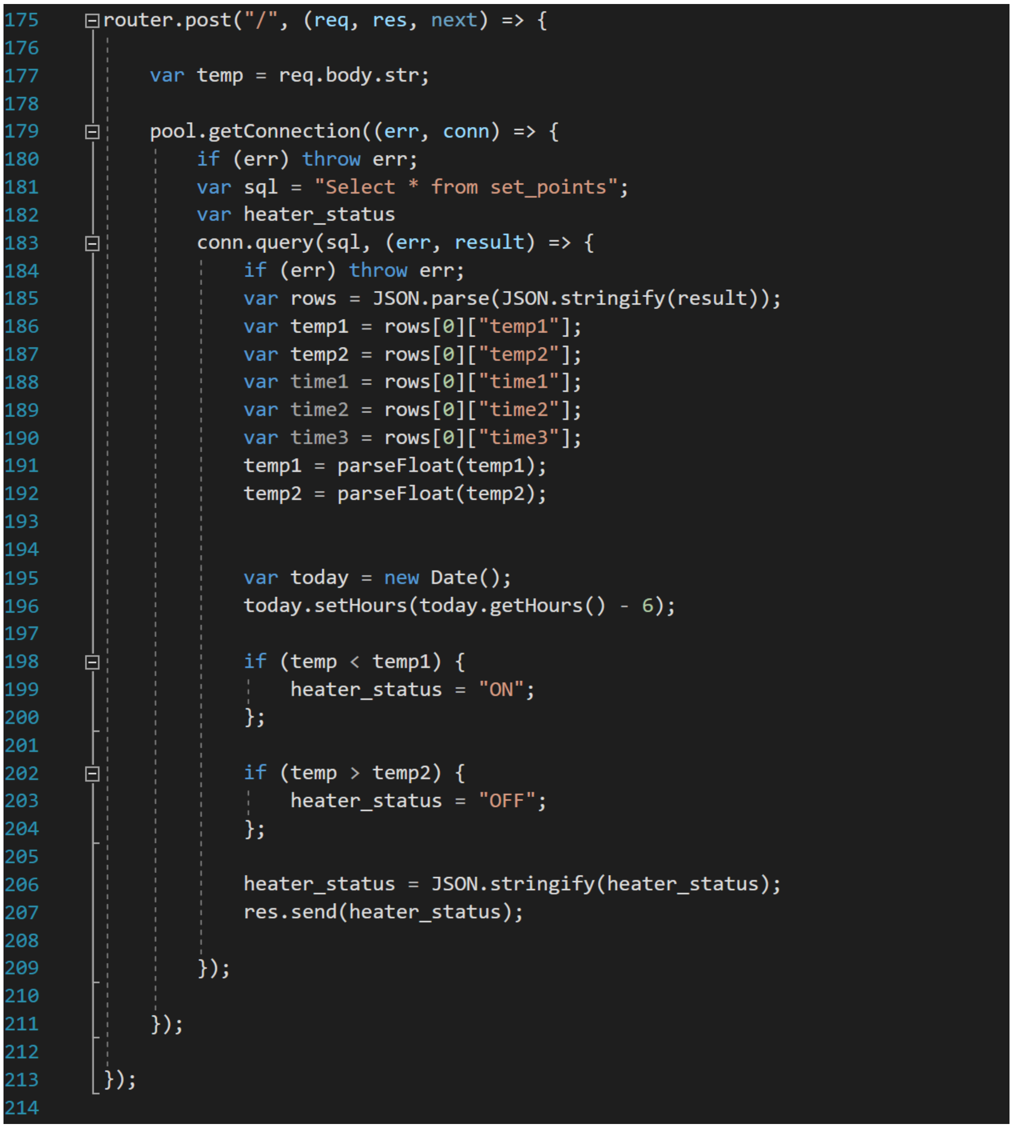


```
body-parser deprecated undefined extended: provide extended option app.js:40:17
Connected to database.
^C
[ec2-user@ip-172-31-5-119 Proj_Repo]$ npm start

> hw@0.0.0 start
> node ./bin/www

body-parser deprecated undefined extended: provide extended option app.js:40:17
Connected to database.
GET /set-points_get 200 46.878 ms - 77
^C
[ec2-user@ip-172-31-5-119 Proj_Repo]$ npm start

> hw@0.0.0 start
> node ./bin/www

body-parser deprecated undefined extended: provide extended option app.js:40:17
Connected to database.
GET /set-points_get 200 40.912 ms - 77
```

After implementing the POST function, the C program and server had far less successful communication. After attempting to fix this, I seemed to have had no communication at all.



```
[ec2-user@ip-172-31-5-119 Proj_Repo]$ npm start

> hw@0.0.0 start
> node ./bin/www

body-parser deprecated undefined extended: provide extended option app.js:40:17
Connected to database.
```



```
router.post("/", (req, res, next) => {

    var temp = req.body.str;

    pool.getConnection((err, conn) => {
        if (err) throw err;
        var sql = "Select * from set_points";
        var heater_status
        conn.query(sql, (err, result) => {
            if (err) throw err;
            var rows = JSON.parse(JSON.stringify(result));
            var temp1 = rows[0]["temp1"];
            var temp2 = rows[0]["temp2"];
            var time1 = rows[0]["time1"];
            var time2 = rows[0]["time2"];
            var time3 = rows[0]["time3"];
            temp1 = parseFloat(temp1);
            temp2 = parseFloat(temp2);


            var today = new Date();
            today.setHours(today.getHours() - 6);

            if (temp < temp1) {
                heater_status = "ON";
            };

            if (temp > temp2) {
                heater_status = "OFF";
            };

            heater_status = JSON.stringify(heater_status);
            res.send(heater_status);

        });

    });

});

```

In index.js, I was attempting to pull the correct time, compare the temperatures, and turn the thermostat on or off based on comparing the current temperature in the provided temp file and the temperatures in my database table *set_points*.