Assignment 22: Understand language - 10422030

Question:

So far in this lesson you have trained a model to understand setting a timer. Another useful feature is cancelling a timer - maybe your bread is ready and can be taken out of the oven before the timer is elapsed.

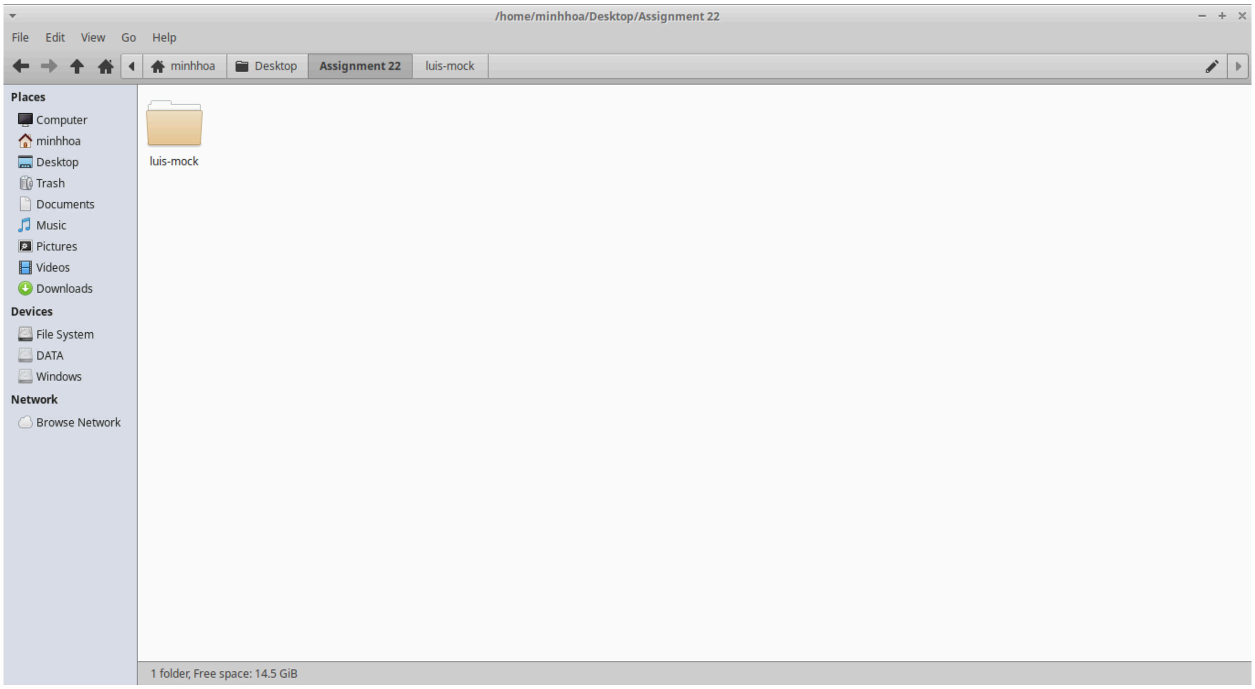
Add a new intent to your LUIS app to cancel the timer. It won't need any entities, but will need some example sentences. Handle this in your serverless code if it is the top intent, logging that the intent was recognized and returning an appropriate response.

Answer:

- In this Assignment, I will create a simulated LUIS app by using a .json file. Because I can’t access the Azure IoT so I will run this Assignment on a local machine.

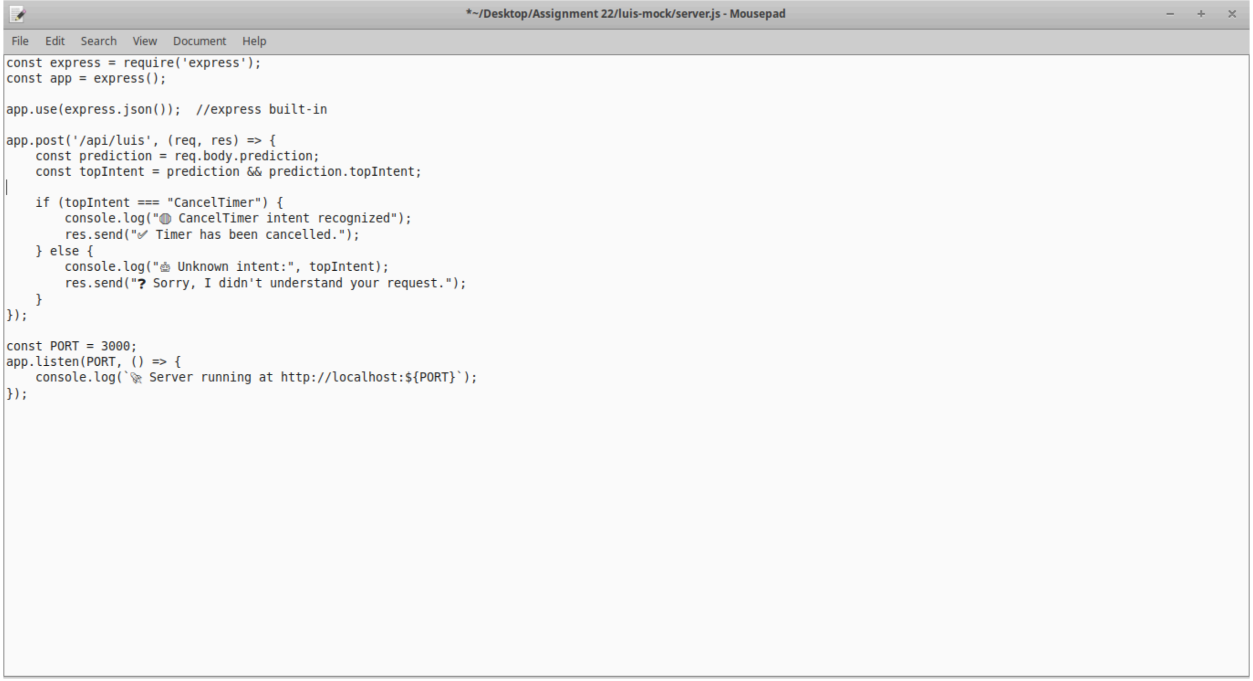
1. Create a folder for Assignment.

- We need to create a folder to store the server.js file, and mock_cancel_timer.json.



2. Create a server.js file.

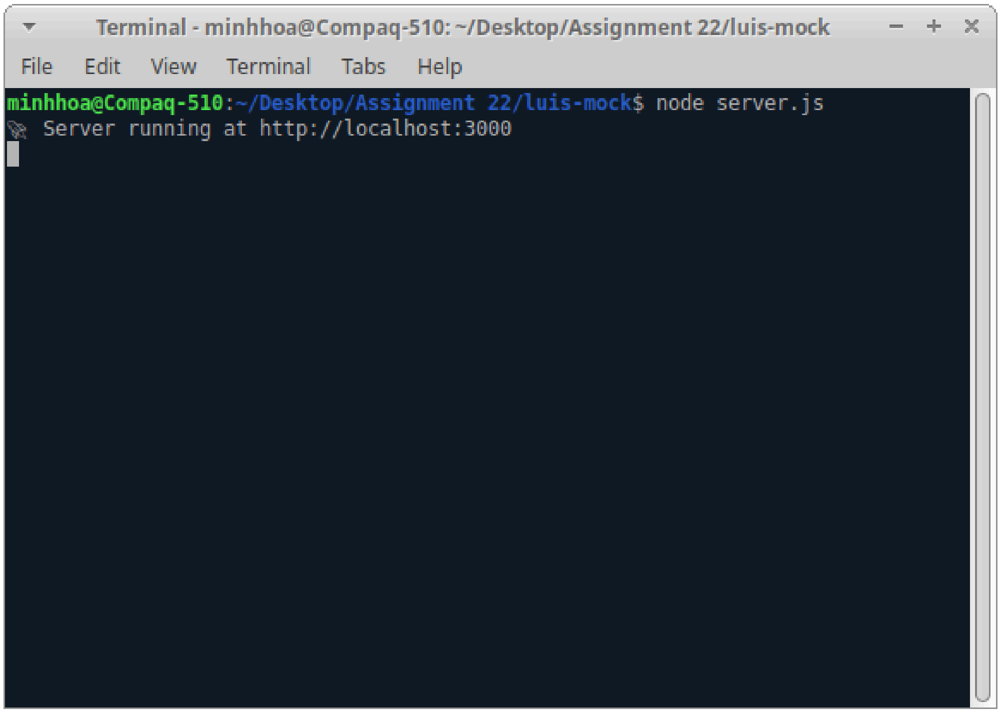
- This file can help us simulate the server to process intent.


*~/Desktop/Assignment 22/luis-mock/server.js - Mousepad
File Edit Search View Document Help

```
const express = require('express');
const app = express();

app.use(express.json());  //express built-in

app.post('/api/luis', (req, res) => {
    const prediction = req.body.prediction;
    const topIntent = prediction && prediction.topIntent;

    if (topIntent === "CancelTimer") {
        console.log("🛑 CancelTimer intent recognized");
        res.send("✔ Timer has been cancelled.");
    } else {
        console.log("🤖 Unknown intent:", topIntent);
        res.send("? Sorry, I didn't understand your request.");
    }
});

const PORT = 3000;
app.listen(PORT, () => {
    console.log(`🚀 Server running at http://localhost:${PORT}`);
});
```

3. Run the server.

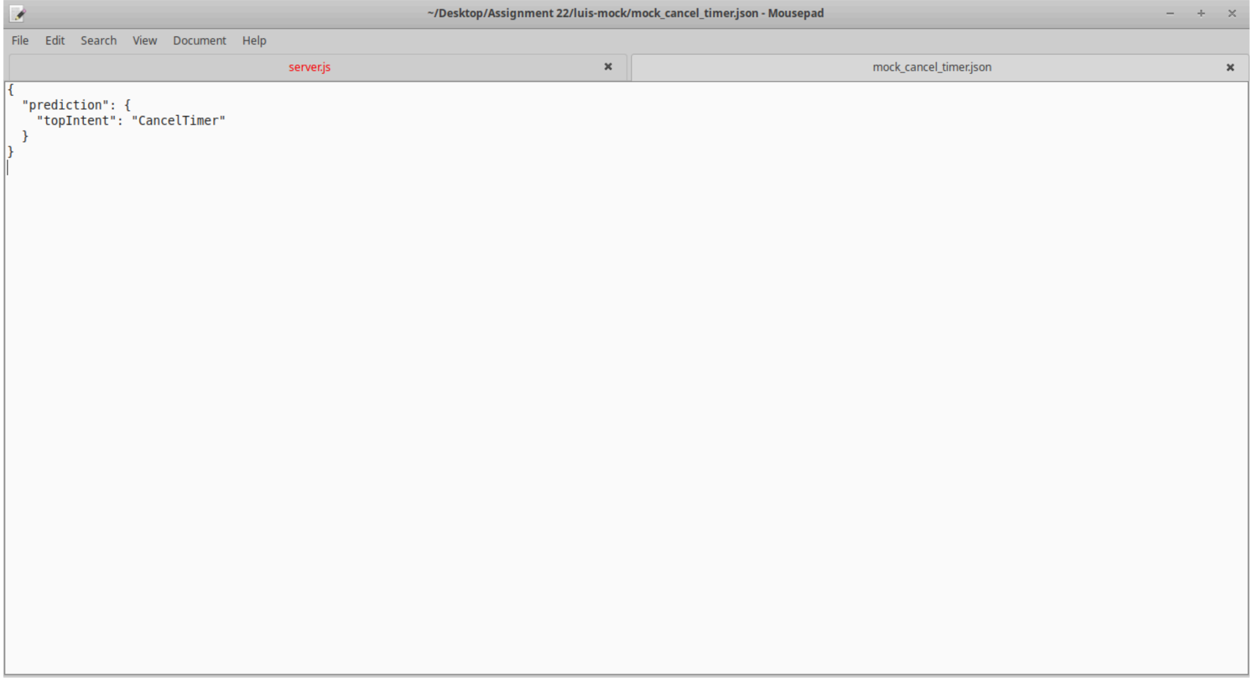
- After created the server.js file, we can run it by the following code: node server.js


Terminal - minhhoa@Compaq-510: ~/Desktop/Assignment 22/luis-mock
File Edit View Terminal Tabs Help

```
minhhoa@Compaq-510:~/Desktop/Assignment 22/luis-mock$ node server.js
🚀 Server running at http://localhost:3000
```

4. Create the mock_cancel_timer.json file.

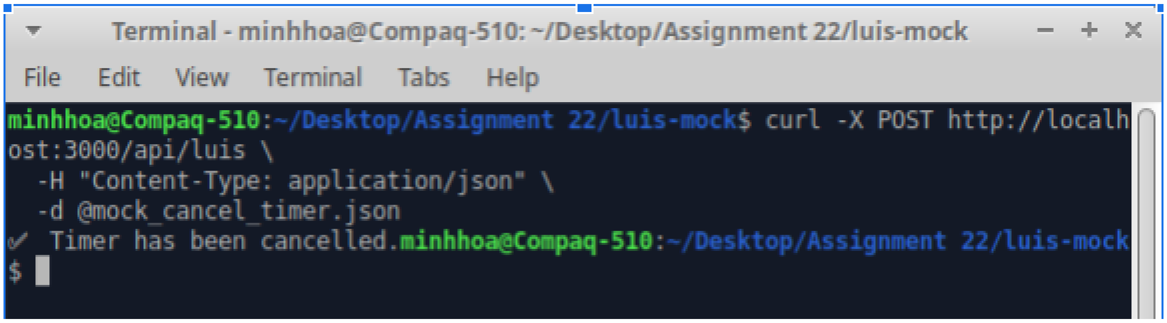
- We need this file to control and connect to the simulated server.


~/Desktop/Assignment 22/luis-mock/mock_cancel_timer.json - Mousepad
File Edit Search View Document Help
server.js
mock_cancel_timer.json

```
{
  "prediction": {
    "topIntent": "CancelTimer"
  }
}
```

5. Run the code to test.

- After we created and run the simulated server, we need to run the code to check that the mock_cancel_timer.json file can work with server.


Terminal - minhhoa@Compaq-510: ~/Desktop/Assignment 22/luis-mock
File Edit View Terminal Tabs Help

```
minhhoa@Compaq-510:~/Desktop/Assignment 22/luis-mock$ curl -X POST http://localh
ost:3000/api/luis \
  -H "Content-Type: application/json" \
  -d @mock_cancel_timer.json
✔ Timer has been cancelled.minhhoa@Compaq-510:~/Desktop/Assignment 22/luis-mock
$ 
```

- We can see that the code reponse: Timer has been cancelled. That mean we are successful in doing this assignment.

Conclusion

In this Assignment, I have:

+ Created a "LUIS mock app" using JSON.

+ Wrote a simple serverless function (Node.js).

+ Send mock data from the command line using curl.

+ Checked the response and logs.

=> All done and tested successfully