# Implement Endpoints Through Theia Lab

Skills Network

## 1. Establishing Endpoints using Node.js for Retrieving Dealership Information

1. Clone your repository and change the directory to function using the following command:

```
cd agfzb-CloudAppDevelopment_Capstone/functions
```

Copied!

2. Create a **"get-dealership.js"** file inside the "functions" folder for Node.js endpoints. Open the "get-dealership.js" file and copy and paste the provided Node.js code below, making sure to include your Cloudant URL, which you can obtain from the Cloudant service's credentials section.

**get-dealership.js :**

```
const express = require('express');
const app = express();
const port = process.env.PORT || 3000;
const Cloudant = require('@cloudant/cloudant');

```

```
// Initialize Cloudant connection with IAM authentication
async function dbCloudantConnect() {
    try {
        const cloudant = Cloudant({
            plugins: { iamauth: { iamApiKey: '' } }, // Replace with your IAM API key
            url: '', // Replace with your Cloudant URL
        });

        const db = cloudant.use('dealerships');
        console.info('Connect success! Connected to DB');
        return db;
    } catch (err) {
        console.error('Connect failure: ' + err.message + ' for Cloudant DB');
        throw err;
    }
}

let db;

(async () => {
    db = await dbCloudantConnect();
})();

app.use(express.json());

// Define a route to get all dealerships with optional state and ID filters
app.get('/dealerships/get', (req, res) => {
    const { state, id } = req.query;

    // Create a selector object based on query parameters
    const selector = {};
    if (state) {
        selector.state = state;
    }

    if (id) {
        selector.id = parseInt(id); // Filter by "id" with a value of 1
    }

    const queryOptions = {
        selector,
        limit: 10, // Limit the number of documents returned to 10
    };

    db.find(queryOptions, (err, body) => {
        if (err) {
            console.error('Error fetching dealerships:', err);
            res.status(500).json({ error: 'An error occurred while fetching dealerships.' });
        } else {
            const dealerships = body.docs;
            res.json(dealerships);
        }
    });
});

app.listen(port, () => {
    console.log(`Server is running on port ${port}`);
});
```

Copied!

3. Initialize the Node.js project and install the required packages:

```
npm init -y
npm install -s @cloudant/cloudant
npm install express
```

Copied!

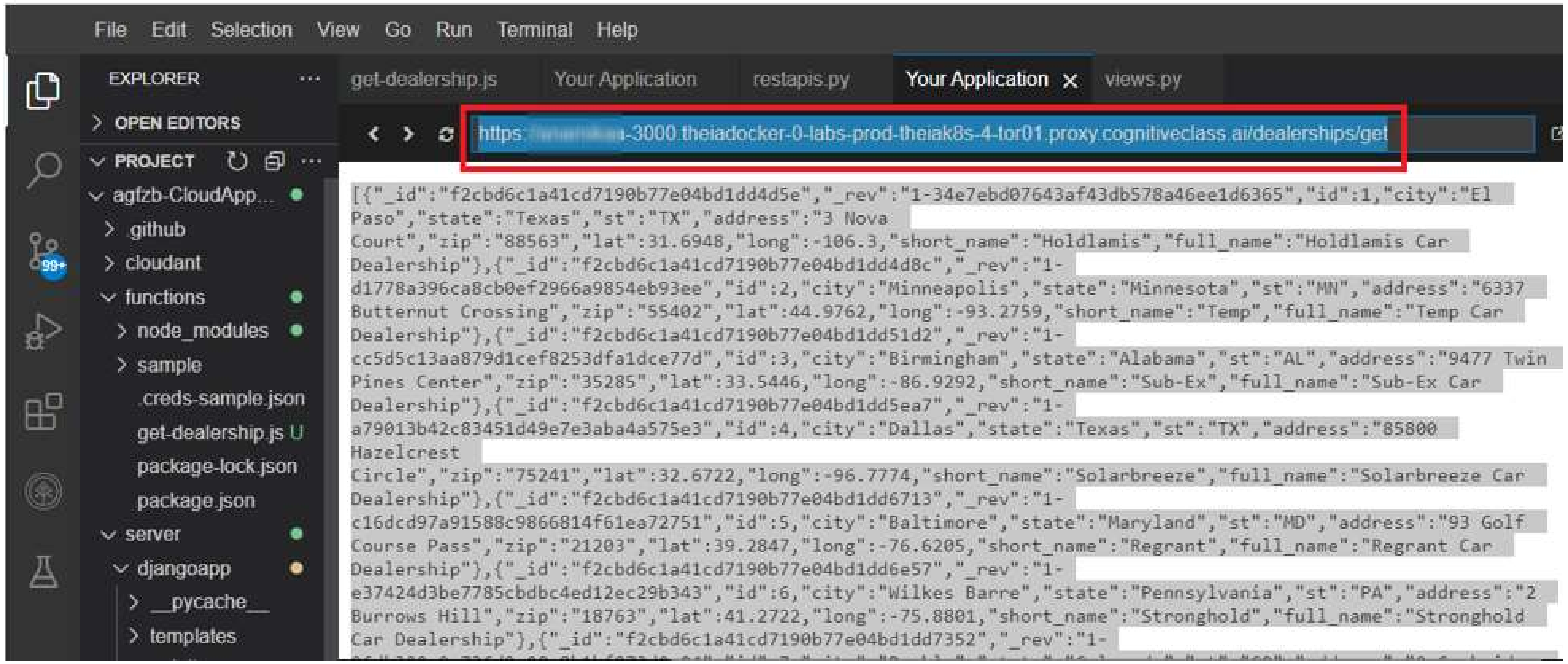
4. Launch the Node.js server for the get-dealership endpoint:

```
node get-dealership.js
```

Copied!

5. Launch the application on port 3000

6. Once the application is launched on port 3000, append **"dealerships/get"** to the end of the URL. This will grant you access to the data from Cloudant, as depicted in the screenshot below.



https[redacted]-3000.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/dealerships/get

```
[{"_id":"f2cbd6c1a41cd7190b77e04bd1dd4d5e","_rev":"1-34e7ebd07643af43db578a46ee1d6365","id":1,"city":"El
Paso","state":"Texas","st":"TX","address":"3 Nova
Court","zip":"88563","lat":31.6948,"long":-106.3,"short_name":"Holdlamis","full_name":"Holdlamis Car
Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd4d8c","_rev":"1-
d1778a396ca8cb0ef2966a9854eb93ee","id":2,"city":"Minneapolis","state":"Minnesota","st":"MN","address":"6337
Butternut Crossing","zip":"55402","lat":44.9762,"long":-93.2759,"short_name":"Temp","full_name":"Temp Car
Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd51d2","_rev":"1-
cc5d5c13aa879d1cef8253dfa1dce77d","id":3,"city":"Birmingham","state":"Alabama","st":"AL","address":"9477 Twin
Pines Center","zip":"35285","lat":33.5446,"long":-86.9292,"short_name":"Sub-Ex","full_name":"Sub-Ex Car
Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd5ea7","_rev":"1-
a79013b42c83451d49e7e3aba4a575e3","id":4,"city":"Dallas","state":"Texas","st":"TX","address":"85800
Hazelcrest
Circle","zip":"75241","lat":32.6722,"long":-96.7774,"short_name":"Solarbreeze","full_name":"Solarbreeze Car
Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd6713","_rev":"1-
c16dcd97a91588c9866814f61ea72751","id":5,"city":"Baltimore","state":"Maryland","st":"MD","address":"93 Golf
Course Pass","zip":"21203","lat":39.2847,"long":-76.6205,"short_name":"Regrant","full_name":"Regrant Car
Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd6e57","_rev":"1-
e37424d3be7785cbdbc4ed12ec29b343","id":6,"city":"Wilkes Barre","state":"Pennsylvania","st":"PA","address":"2
Burrows Hill","zip":"18763","lat":41.2722,"long":-75.8801,"short_name":"Stronghold","full_name":"Stronghold
Car Dealership"},{"_id":"f2cbd6c1a41cd7190b77e04bd1dd7352","_rev":"1-
```

**Take a screenshot of this step and ensure to save it for your final assignment. Additionally, remember to save the URL.**

NOTE: Once you have finished all the Hands-on Labs up to week 4, you can return and proceed with the following steps:

7. Copy this URL and proceed to open the views.py file. Replace the existing URL for get-dealership with the newly copied endpoint URL in the get_dealerships function.
8. Now, open the restapis.py file and in the 'get_dealers_from_cf' function, replace this previous line, which appears as follows:

   **dealer_doc = dealer["doc"]**

   with this new line below:

```
dealer_doc = dealer
```
Copied!

   This will update the way the **dealer_doc** variable retrieves the content from the dealer object.

9. Ensure that your previous Node.js server is still running in the background, and then proceed to open a new terminal window.
10. Change the current directory to the server by executing the command:

```
cd agfzb-CloudAppDevelopment_Capstone/server
```

Copied!

11. Install the required Python packages using the following command:

```
python3 -m pip install -U -r requirements.txt
```

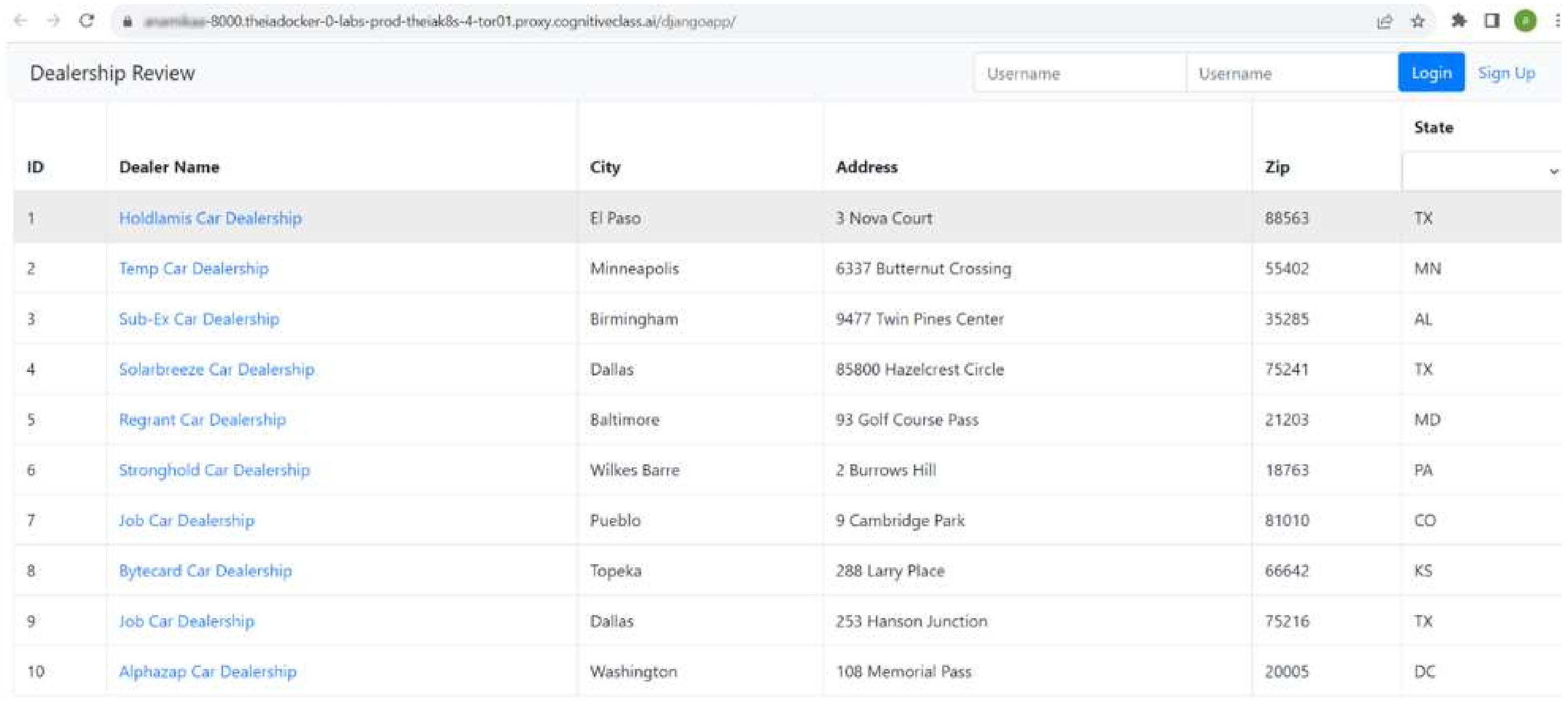
Copied!

12. Finally, run the server with the following command:

```
python3 manage.py runserver
```

Copied!

13. Launch your web browser and access the application on port 8000. As a result, you will be able to view the user interface displaying all the dealership data as shown in the below screenshot.



[redacted]-8000.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/djangoapp/

Dealership Review

Username Username Login Sign Up

| ID | Dealer Name | City | Address | Zip | State |
|---|---|---|---|---|---|
| 1 | Holdlamis Car Dealership | El Paso | 3 Nova Court | 88563 | TX |
| 2 | Temp Car Dealership | Minneapolis | 6337 Butternut Crossing | 55402 | MN |
| 3 | Sub-Ex Car Dealership | Birmingham | 9477 Twin Pines Center | 35285 | AL |
| 4 | Solarbreeze Car Dealership | Dallas | 85800 Hazelcrest Circle | 75241 | TX |
| 5 | Regrant Car Dealership | Baltimore | 93 Golf Course Pass | 21203 | MD |
| 6 | Stronghold Car Dealership | Wilkes Barre | 2 Burrows Hill | 18763 | PA |
| 7 | Job Car Dealership | Pueblo | 9 Cambridge Park | 81010 | CO |
| 8 | Bytecard Car Dealership | Topeka | 288 Larry Place | 66642 | KS |
| 9 | Job Car Dealership | Dallas | 253 Hanson Junction | 75216 | TX |
| 10 | Alphazap Car Dealership | Washington | 108 Memorial Pass | 20005 | DC |

## 2. Establishing Endpoints in Python for Retrieving Reviews and Posting Reviews.

1. Begin by creating a new file named “reviews.py” within the “functions” directory. Copy and paste the provided code into this file. Ensure to replace the API key, username, and URL with your specific Cloudant service credentials.

```
from cloudant.client import Cloudant
from cloudant.query import Query
from flask import Flask, jsonify, request
import atexit

#Add your Cloudant service credentials here
cloudant_username = ''
cloudant_api_key = ''
cloudant_url = ''
client = Cloudant.iam(cloudant_username, cloudant_api_key, connect=True, url=cloudant_url)

session = client.session()
print('Databases:', client.all_dbs())

db = client['reviews']

app = Flask(__name__)

@app.route('/api/get_reviews', methods=['GET'])
def get_reviews():
    dealership_id = request.args.get('id')

    # Check if "id" parameter is missing
    if dealership_id is None:
        return jsonify({"error": "Missing 'id' parameter in the URL"}), 400

    # Convert the "id" parameter to an integer (assuming "id" should be an integer)
    try:
        dealership_id = int(dealership_id)
    except ValueError:
        return jsonify({"error": "'id' parameter must be an integer"}), 400

    # Define the query based on the 'dealership' ID
    selector = {
        'dealership': dealership_id
    }

    # Execute the query using the query method
    result = db.get_query_result(selector)

    # Create a list to store the documents
    data_list = []

    # Iterate through the results and add documents to the list
    for doc in result:
        data_list.append(doc)

    # Return the data as JSON
    return jsonify(data_list)


@app.route('/api/post_review', methods=['POST'])
def post_review():
    if not request.json:
        abort(400, description='Invalid JSON data')

    # Extract review data from the request JSON
    review_data = request.json

    # Validate that the required fields are present in the review data
    required_fields = ['id', 'name', 'dealership', 'review', 'purchase', 'purchase_date', 'car_make', 'car_model', 'car_year']
    for field in required_fields:
        if field not in review_data:
            abort(400, description=f'Missing required field: {field}')

    # Save the review data as a new document in the Cloudant database
    db.create_document(review_data)

    return jsonify({"message": "Review posted successfully"}), 201

if __name__ == '__main__':
    app.run(debug=True)
```

Copied!

2. Install the Cloudant library by executing the following commands:

```
python3.8 -m pip install --upgrade pip
pip3.8 install Cloudant
```

Copied!

3. Run the Python script “reviews.py” using the command:

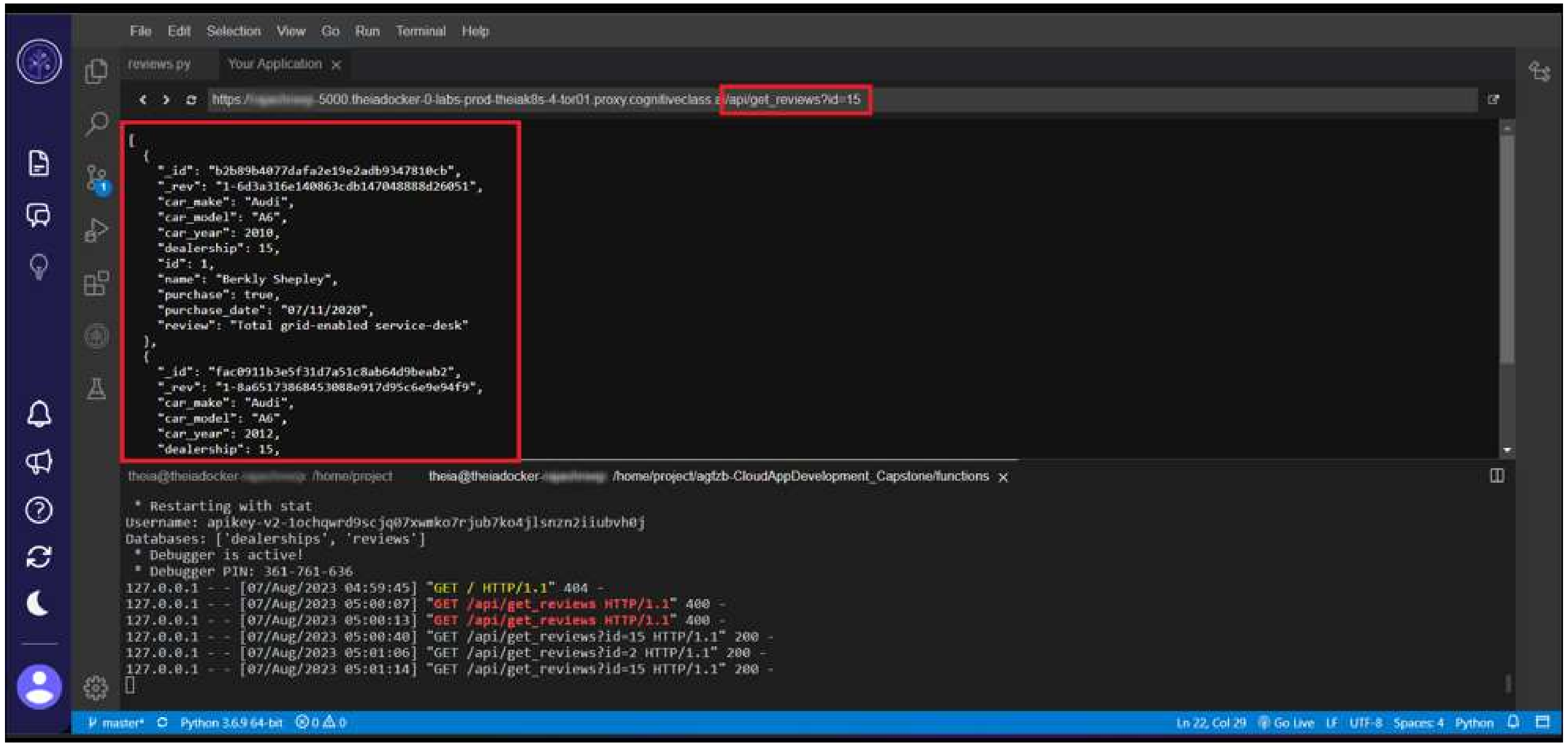
```
python3.8 reviews.py
```

Copied!

This will start the server, allowing you to launch your application on port 5000.

4. To access reviews for a specific ID, append **“api/get_reviews?id=15”** to the URL. You can replace “15” with any desired ID. This will grant you access to the respective review.

**Please find the attached screenshot for your convenience.**


File Edit Selection View Go Run Terminal Help
reviews.py
Your Application
https://[redacted]-5000.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/api/get_reviews?id=15
theia@theiadocker-[redacted]: /home/project
theia@theiadocker-[redacted]: /home/project/agtzb-CloudAppDevelopment_Capstone/functions
master* Python 3.6.9 64-bit ⊗ 0 ⚠ 0
Ln 22, Col 29 Go Live LF UTF-8 Spaces: 4 Python

```
[
  {
    "_id": "b2b89b4077dafa2e19e2adb9347810cb",
    "_rev": "1-6d3a316e140863cdb147048888d26051",
    "car_make": "Audi",
    "car_model": "A6",
    "car_year": 2010,
    "dealership": 15,
    "id": 1,
    "name": "Berkly Shepley",
    "purchase": true,
    "purchase_date": "07/11/2020",
    "review": "Total grid-enabled service-desk"
  },
  {
    "_id": "fac0911b3e5f31d7a51c8ab64d9beab2",
    "_rev": "1-8a65173868453088e917d95c6e9e94f9",
    "car_make": "Audi",
    "car_model": "A6",
    "car_year": 2012,
    "dealership": 15,
```

```
 * Restarting with stat
Username: apikey-v2-1ochqwrd9scjq07xwmko7rjub7ko4jlsnzn2iiubvh0j
Databases: ['dealerships', 'reviews']
 * Debugger is active!
 * Debugger PIN: 361-761-636
127.0.0.1 - - [07/Aug/2023 04:59:45] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [07/Aug/2023 05:00:07] "GET /api/get_reviews HTTP/1.1" 400 -
127.0.0.1 - - [07/Aug/2023 05:00:13] "GET /api/get_reviews HTTP/1.1" 400 -
127.0.0.1 - - [07/Aug/2023 05:00:40] "GET /api/get_reviews?id=15 HTTP/1.1" 200 -
127.0.0.1 - - [07/Aug/2023 05:01:06] "GET /api/get_reviews?id=2 HTTP/1.1" 200 -
127.0.0.1 - - [07/Aug/2023 05:01:14] "GET /api/get_reviews?id=15 HTTP/1.1" 200 -
```

**Take a screenshot of this step and ensure to save it for your final assignment. Additionally, remember to save the URL.**

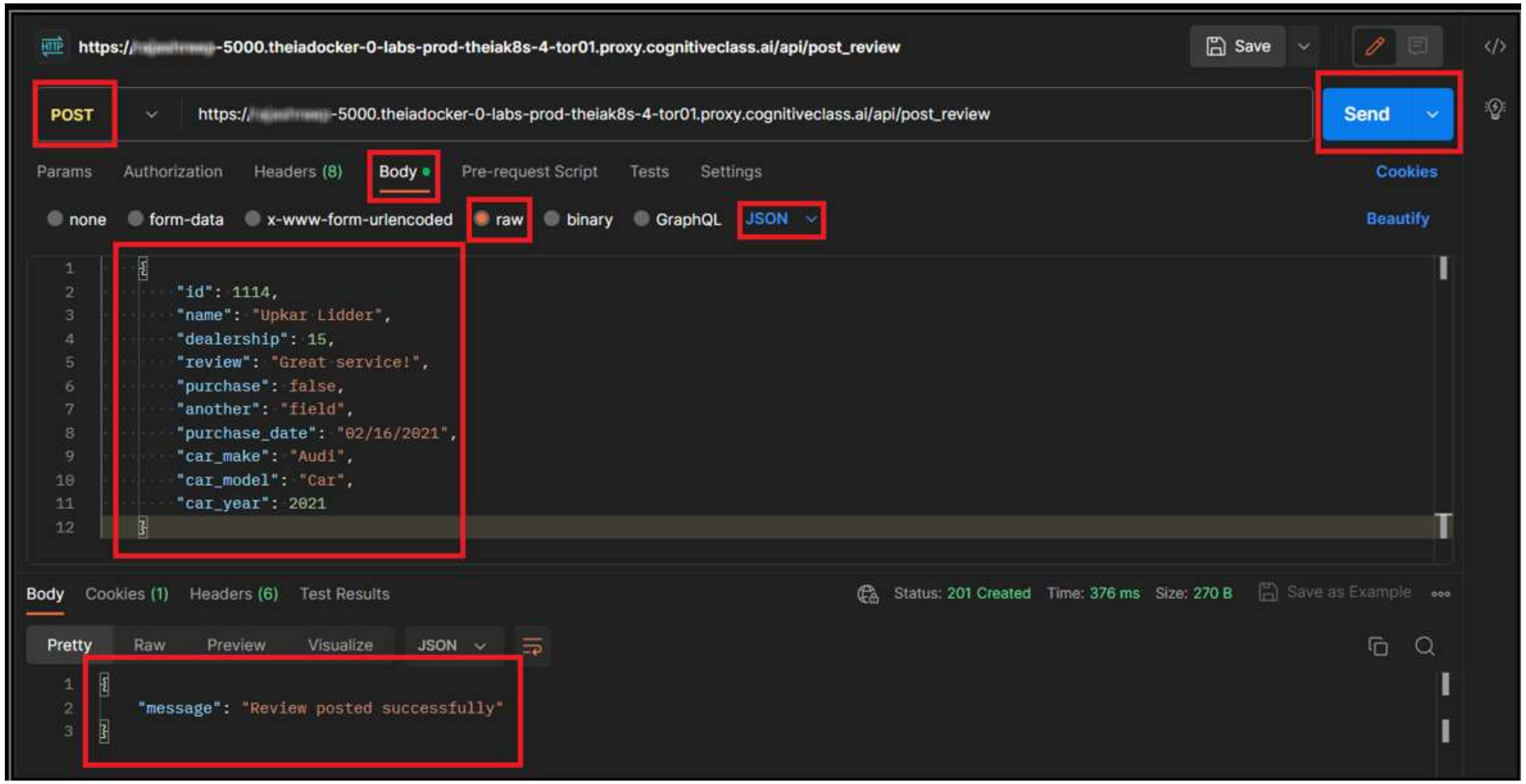
5. For posting a new review, open Postman. Paste the URL you obtained after launching the server and then append **“api/post_review”** to it. Set the method as **“POST”** In Postman, select **“Body”** then choose **“raw”** and **“json”** to send the data. Input the provided data and click the **“Send”** button.

```
{
        "id": 1114,
        "name": "Upkar Lidder",
        "dealership": 15,
        "review": "Great service!",
        "purchase": false,
        "another": "field",
        "purchase_date": "02/16/2021",
        "car_make": "Audi",
        "car_model": "Car",
        "car_year": 2021
    }
```

Copied!

6. Upon a successful request, you'll receive a confirmation message stating **"Review posted successfully."**

**Please find the attached screenshot for your convenience.**



https://[redacted]-5000.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/api/post_review

POST https://[redacted]-5000.theiadocker-0-labs-prod-theiak8s-4-tor01.proxy.cognitiveclass.ai/api/post_review

```
{
    "id": 1114,
    "name": "Upkar Lidder",
    "dealership": 15,
    "review": "Great service!",
    "purchase": false,
    "another": "field",
    "purchase_date": "02/16/2021",
    "car_make": "Audi",
    "car_model": "Car",
    "car_year": 2021
}
```

Status: 201 Created Time: 376 ms Size: 270 B

```
{
    "message": "Review posted successfully"
}
```

**Take a screenshot of this step and ensure to save it for your final assignment. Additionally, remember to save the URL.**

*Note: You have the option to keep the servers running in the background and utilize these URLs within your "view.py" file, just as you did for the Node.js endpoints in the earlier instructions, to launch your Django application.*

## Congratulations! You have completed this lab

## Changelog

| Date | Version | Changed by | Change Description |
|---|---|---|---|
| | 0.1 | | Initial version created |
| 11-10-2023 | 0.2 | K Sundararajan | Updated code for `get-dealership.js` |