Project 3 - RESTful Server

Re-submit Assignment

Due Dec 10 by 5pm **Points** 100 **Submitting** a text entry box or a website url
Available Nov 8 at 5pm - Dec 10 at 5pm about 1 month

Assignment 3

RESTful Web Server

Task - PART 1

Create a RESTful web server for St. Paul crime data. Your server should implement a number of API routes relating to the data.

NOTE: you are allowed to use any Node.js modules (built-in, installed via npm, or written yourself) to help develop your RESTful web server.

About the Data Set

St. Paul Crime Database:

I have downloaded data from the St. Paul public dataset located at **https://information.stpaul.gov/Public-Safety/Crime-Incident-Report-Dataset/gppb-g9cg (https://information.stpaul.gov/Public-Safety/Crime-Incident-Report-Dataset/gppb-g9cg)** and stored them in a SQLite3 database.

The database has 3 tables as follows:

- Codes:
  - code (INTEGER) - crime incident type numeric code
  - incident_type (TEXT) - crime incident type description
- Neighborhoods:
  - neighborhood_number (INTEGER) - neighborhood id
  - neighborhood_name (TEXT) - neighborhood name
- Incidents:
  - case_number (TEXT): unique id from crime case
  - date_time (DATETIME): date and time when incident took place
  - code (INTEGER): crime incident type numeric code
  - incident (TEXT): crime incident description (more specific than incident_type)
  - police_grid (INTEGER): police grid number where incident occurred
  - neighborhood_number (INTEGER): neighborhood id where incident occurred

- block (TEXT): approximate address where incident occurred

RESTful Web Server (40 pts)

Implement the following to earn 30/40 points (grade: C)

- Package.json
  - Fill out the author and contributors sections in package.json (author should be whoever's GitHub account is used to host the code, contributors should be all group members)
  - Fill out the URL of the repository
  - Ensure all used modules downloaded via NPM are in the dependencies object
- Add the following routes for your API
  - GET /codes
    - Return JSON object with list of codes and their corresponding incident type (ordered by code)
    - Example:

```
{
  "110": "Murder, Non Negligent Manslaughter",
  "120": "Murder, Manslaughter By Negligence",
  "210": "Rape, By Force",
  "220": "Rape, Attempt",
  "300": "Robbery",
  "311": "Robbery, Highway, Firearm",
  "312": "Robbery, Highway, Knife or Cutting Instrument",
  "313": "Robbery, Highway, Other Dangerous Weapons",
  "314": "Robbery, Highway, By Strong Arm",
  ...
}
```

  - GET /neighborhoods
    - Return JSON object with list of neighborhood ids and their corresponding neighborhood name (ordered by neighborhood number)
    - Example:

```
{
  "1": "Conway/Battlecreek/Highwood",
  "2": "Greater East Side",
  "3": "West Side",
  "4": "Dayton's Bluff",
  "5": "Payne/Phalen",
  "6": "North End",
  "7": "Thomas/Dale(Frogtown)",
  "8": "Summit/University",
  "9": "West Seventh",
  "10": "Como",
  "11": "Hamline/Midway",
  "12": "St. Anthony",
  "13": "Union Park",
  "14": "Macalester-Groveland",
  "15": "Highland",
  "16": "Summit Hill",
```

```
  "17": "Capitol River"
}
```

- GET /incidents
  - Return JSON object with list of crime incidents (most recent first). Make date and time separate fields.
  - Example:

```
{
  "19245020": {
    "date": "2019-10-30",
    "time": "23:57:08",
    "code": 9954,
    "incident": "Proactive Police Visit",
    "police_grid": 87,
    "neighborhood_number": 7,
    "block": "THOMAS AV  & VICTORIA"
  },
  "19245016": {
    "date": "2019-10-30",
    "time": "23:53:04",
    "code": 9954,
    "incident": "Proactive Police Visit",
    "police_grid": 87,
    "neighborhood_number": 7,
    "block": "98X UNIVERSITY AV W"
  },
  "19245014": {
    "date": "2019-10-30",
    "time": "23:43:19",
    "code": 700,
    "incident": "Auto Theft",
    "police_grid": 95,
    "neighborhood_number": 4,
    "block": "79X 6 ST E"
  },
  ...
}
```

- PUT /new-incident
  - Upload incident data to be inserted into the SQLite3 database
  - Data fields:
    - case_number
    - date
    - time
    - code
    - incident
    - police_grid
    - neighborhood_number
    - block
  - Note: response should reject (status 500) if the case number already exists in the database

Implement additional features to earn a B or A

- Add the following query options for GET /codes **(2 pts)**
  - code - comma separated list of codes to include in result (e.g. `?code=110,700`). By default all codes should be included.
  - format - json or xml (e.g. `?format=xml`). By default JSON format should be used.
- Add the following query options for GET /neighborhoods **(2 pts)**
  - id - comma separated list of neighborhood numbers to include in result (e.g. `?id=11,14`). By default all neighborhoods should be included.
  - format - json or xml (e.g. `?format=xml`). By default JSON format should be used.
- Add the following query options for GET /incidents **(6 pts)**
  - start_date - first date to include in results (e.g. `?start_date=09-01-2019`)
  - end_date - last date to include in results (e.g. `?end_date=10-31-2019`)
  - code - comma separated list of codes to include in result (e.g. `?code=110,700`). By default all codes should be included.
  - grid - comma separated list of police grid numbers to include in result (e.g. `?grid=38,65`). By default all police grids should be included.
  - neighborhood - comma separated list of neighborhood numbers to include in result (e.g. `?id=11,14`). By default all neighborhoods should be included.
  - limit - maximum number of incidents to include in result (e.g. `?limit=50`). By default the limit should be 10,000.
  - format - json or xml (e.g. `?format=xml`). By default JSON format should be used.

Starter Code

There is no starter code for this project - only the data. Download **stpaul_crime.sqlite3**: database with crime data.

Submission

Code should be saved in a repository on GitHub. Do NOT add your node_modules directory to your repository. This is what package.json is for - it will store which modules you use for your project. In order to submit, you should enter the the project's GitHub URL for the assignment (in Canvas). I will be doing the following to assess your assignment:

1. git clone https://github.com/*<user>*/*<project>*
2. cd *<project>*
3. npm install
4. node server.js

IMPORTANT: Only one group member should submit the GitHub URL. Every member should submit a checklist of what you feel you have accomplished from the rubric above (including who did what), and include

your total expected score. This can be as a text entry submission (if not submitting the URL), or as a comment once you submit the URL.

Groups

Section 1

Matt & Jesse	Jack & Summer	Kristina & Cullen	Mitzi & Mugdha
Nick & Devin	Saleena & Owen	Danielle & Abby	Lani & Gafar
Rachel D. & Jacob J.	Tim & Rachel R.	Kong & Dylan	Lao R. & Leo
Jacob R. & Tucker			

Section 2

Austin & Riley	Ben & Scott	Haoyi & Duke	Trevor & Nick
Galvin & Aaron	Charlie & Andrew	Shijun & Luca	Shiela & Tom
Drew & Alex Y.	George & Khalid	Alex P. & Daniel	Brandon, Salmaan, & Gwen

Deadline

PART 1 has a deadline of Thursday November 21 at 5:00pm.

PART 2 will be due Tuesday, December 10 at 5:00pm.