Project 3 - RESTful Server

Due Nov 30, 2021 by 10pm **Points** 40 **Submitting** a text entry box or a website url
Available after Nov 15, 2021 at 1:35pm

Assignment 3

RESTful Web Server

Task - Note: this is PART 1 of a two part project

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 array with list of codes and their corresponding incident type (ordered by code number)
    - Example:

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

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

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

```
  {"id": 16, "name": "Summit Hill"},
  {"id": 17, "name": "Capitol River"}
]
```

- GET /incidents
  - Return JSON object with list of crime incidents (ordered by date/time). Note date and time should be separate fields.
  - Example:

```
[
  {
    "case_number": "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"
  },
  {
    "case_number": "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"
  },
  {
    "case_number": "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
- DELETE /remove-incident

- Remove data from the SQLite3 database
- Data fields:
  - case_number
- Note: reponse should reject (status 500) if the case number does not exist in the database

Implement additional features to earn a B or A

- Add the following query option 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.
- 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.
- Add the following query options for GET /incidents **(6 pts)**
  - start_date - first date to include in results (e.g. `?start_date=2019-09-01`)
  - end_date - last date to include in results (e.g. `?end_date=2019-10-31`)
  - 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. `?neighborhood=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 1,000. Result should include the N most recent incidents (within specified date range).

Database

The database can be downloaded here (**stpaul_crime.sqlite3** (https://stthomas.instructure.com/courses/46313/files/5069571/download?download_frd=1))

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. Copy my local version of 'stpaul_crime.sqlite3' to the 'db' folder
4. npm install
5. node server.js
6. Perform GET, PUT, DELETE requests using curl

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

Ben F., Grant, & Logan	Antonio, Nolan, & Lucas	Jackson, Brandon B., & Michael
Jack, Zak, & Carter	Ben E., Daniel, & Anshul	Brandon T., Joe, & Aaron
Tseng, Emma, & Elizabeth	Peter, Tonya, & Noah	Zack & Erik

Deadline

This project is due Tuesday, November 30 at 10:00pm.