

# Trazi

## City Population Service

You will be writing a node.js service that allows you to get a cities population by state + city while also letting you set a cities population. This service will be hit hard and needs to return data as fast as possible. So make sure you write this service with response time & throughput as the focus when writing your code.

### Requirements:

- Use Node.js v18
- Written in plain javascript (**no typescript**)
- Running the command `npm install` will install anything necessary to run the service
- Running the command `npm start` will start your service and all required things for your service running.
- Your service will run on port `5555`
- The data must **persist**
- Must have the route `GET http://127.0.0.1:5555/api/population/state/:state/city/:city`
  - This allows a `:state` and a `:city` to be passed in. For example (<http://127.0.0.1:5555/api/population/state/Florida/city/Orlando>)
  - State and city should not be case sensitive
  - Should return back with a 400 if the state / city combo cant be found with a proper error message
  - Should return back with a 200 status and json response if found. For example `{"population": 32423}`
- Must have the route `PUT http://127.0.0.1:5555/api/population/state/:state/city/:city`
  - This allows a `:state` and a `:city` to be passed in. For example (<http://127.0.0.1:5555/api/population/state/Florida/city/Orlando>)
  - State and city should not be case sensitive
  - Body should be plain text that contains just the number to be set as the population
  - Should return back with a 400 if data could not be added and proper error message
  - Should return back a 200 status if the data has updated a state / city that already existed and should return back a 201 if the data was created instead of updated.

## The Data:

The data can be found at

[https://github.com/Trazi-Ventures/sample-data-interview/blob/main/city\\_populations.csv](https://github.com/Trazi-Ventures/sample-data-interview/blob/main/city_populations.csv). In regards to this data's format, schema, how it's stored, etc you are allowed to change it. The only rule is; the data is returned back with the expected population that's in this file. For example We expect that for the request `http://127.0.0.1:5555/api/population/state/Alabama/city/Marion` The population of 3178 is returned. So feel free to store it in any format you want using whatever way you want. Just remember response time and throughput is key here.

## What we are looking for:

- **Fast response time** - The lower you can get this number the better. Hint - Are you sure you're using the fastest REST framework for node?
- **High throughput** - How much concurrent requests/users can your service handle
- **Minimal use of 3rd party modules** - Lots of 3rd party modules don't have speed as their primary focus, so ask yourself if using a package for something if it's really helping you meet your goal.
- **Optimized javascript code** - How well you utilize javascript to meet your goal instead of being hindered by some of its conveniences.
- **Structured code** - Is it all in a giant blob file or is it well structured for ease of development and future expansion.