# API Introduction and Setup an API Server with MySQL and Express

## Overview

In this project you will need to create a REST api and display the data in a jQuery Datatable.

Part 1 – Get my code to work

This project is to setup a MySQL database and load it with cities data. Then you need to create an express server to serve your static html files and send JSON data to your homepage to render a data table. This project cannot run on GitHub pages because it has a server. You will be using Docker to run your MySQL server, express is your API server, and express serves your static pages. You will not use Browser Sync and I removed it for this assignment. BrowserSync is good to know; however, in this assignment the main focus is the data from the server, and I set it up to use "NodeMon" to reload the front end and the server when files are changes. Essentially, browser sync can't automatically request data from the backend and that's what we need for this. You can find steps to setup my project at the end of this document.

Part 2 – Build your own REST api with one of the following CSV files as the source:

https://people.sc.fsu.edu/~jburkardt/data/csv/csv.html

Create the REST API and the datatable view of that data.

Use this to convert 100 records of the CSV file to an SQL insert:
https://sqlizer.io/#/

https://codebeautify.org/csv-to-sql-converter
this one works very well and doesn't seem to have a limit, but I haven't tried it for the project.

You can view and edit your CSV data in excel before you convert it, so you limit it to 100 records and can rename any fields, just save it as a CSV file.

## How to Submit your Project

Create a new word document that includes the following:

1. A link to your GitHub repository for a NEW project
2. Submit screen shots of the following:
   a. *The project showing that you can view **YOUR** data tables in WebStorm*
   b. ***YOUR** project running correctly in the browser displaying a data table*
   c. *A screen shot of Postman and **YOUR** data returned from the project.*
   d. *A screen shot of the browser console showing the XHR network request made from the browser when the page loads.*

Readings and Lectures

- Video for the Project: https://youtu.be/GEkrDLZLvY0
- What is an API https://www.youtube.com/watch?v=E1Xk5dk3Abc <- It's another video I made but I don't want to record the exact same information for JavaScript, this is general information that you should know about what we are doing and isn't specific to JavaScript.
- Part 2 of What is an API: https://www.youtube.com/watch?v=LGNkFTE3DI8
- What is AJAX? https://skillcrush.com/blog/what-is-ajax/

# Project Setup Instructions

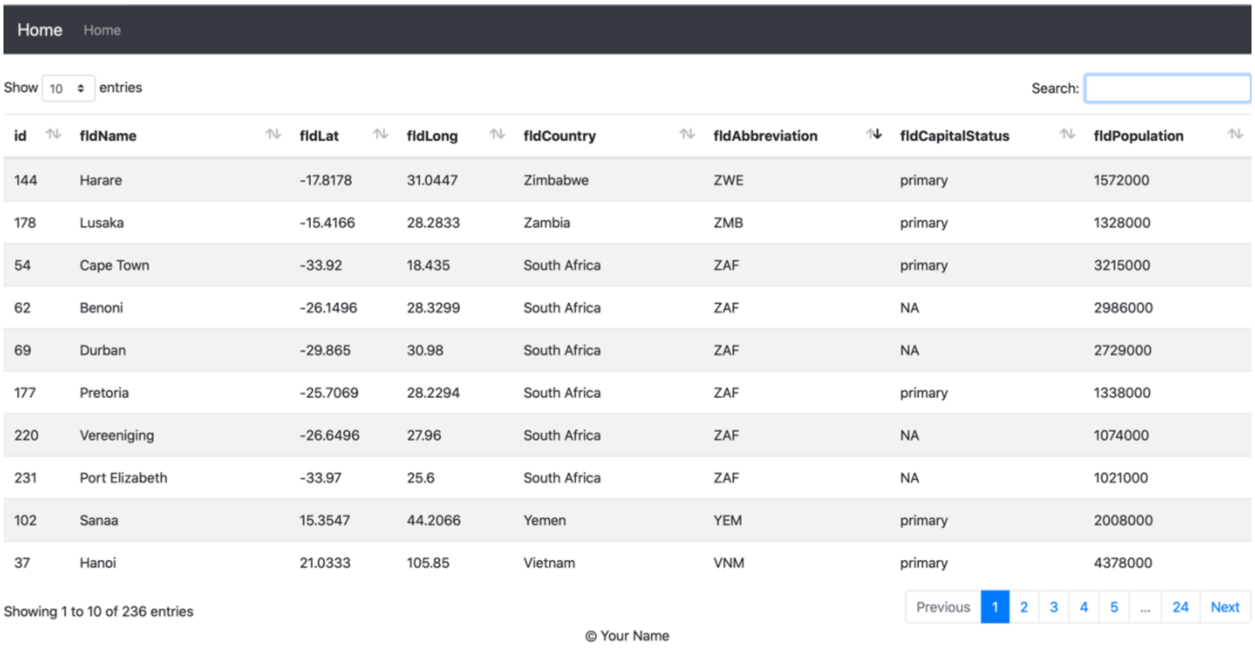
Libraries Used:

https://datatables.net
https://expressjs.com
https://jquery.com
https://getbootstrap.com
https://nodejs.org/en/
https://www.npmjs.com
https://www.mysql.com

Technology Used:

https://www.docker.com
https://www.postman.com

Working Assignment Homepage:



| id | fldName | fldLat | fldLong | fldCountry | fldAbbreviation | fldCapitalStatus | fldPopulation |
|---|---|---|---|---|---|---|---|
| 144 | Harare | -17.8178 | 31.0447 | Zimbabwe | ZWE | primary | 1572000 |
| 178 | Lusaka | -15.4166 | 28.2833 | Zambia | ZMB | primary | 1328000 |
| 54 | Cape Town | -33.92 | 18.435 | South Africa | ZAF | primary | 3215000 |
| 62 | Benoni | -26.1496 | 28.3299 | South Africa | ZAF | NA | 2986000 |
| 69 | Durban | -29.865 | 30.98 | South Africa | ZAF | NA | 2729000 |
| 177 | Pretoria | -25.7069 | 28.2294 | South Africa | ZAF | primary | 1338000 |
| 220 | Vereeniging | -26.6496 | 27.96 | South Africa | ZAF | NA | 1074000 |
| 231 | Port Elizabeth | -33.97 | 25.6 | South Africa | ZAF | NA | 1021000 |
| 102 | Sanaa | 15.3547 | 44.2066 | Yemen | YEM | primary | 2008000 |
| 37 | Hanoi | 21.0333 | 105.85 | Vietnam | VNM | primary | 4378000 |

Showing 1 to 10 of 236 entries

© Your Name

The completed code for this project can be found here:

https://github.com/kaw393939/is219Project2/tree/assignment3

*Pre-Requisites*

1. *WebStorm Installed*
2. *Node.js Installed*
3. *Docker Installed*

4. *Postman Installed*

## Project Steps

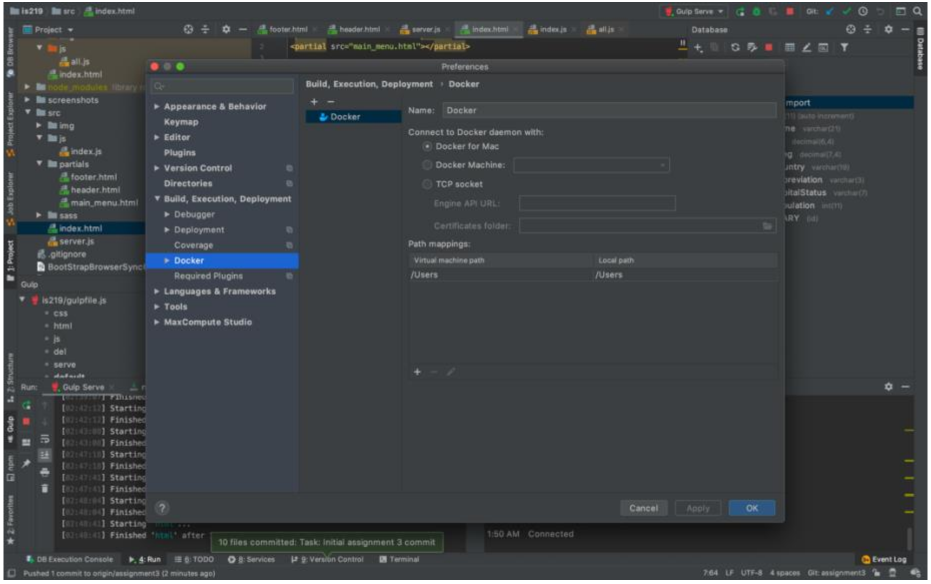
Step 1 – Clone the project to WebStorm

https://github.com/kaw393939/is219Project2

Step 2 – Switch branches to assignment 3 branch

Step 3 – Run npm install in the terminal of the project

Step 4 – Make sure docker is connected in your build preferences in webstorm. For PC it will be different. You need to choose the TCP socket and make sure it is exposed in the docker configuration.



Preferences

Build, Execution, Deployment › Docker

Name: Docker

Connect to Docker daemon with:
- ● Docker for Mac
- ○ Docker Machine:
- ○ TCP socket
  - Engine API URL:
  - Certificates folder:

Path mappings:

| Virtual machine path | Local path |
|---|---|
| /Users | /Users |

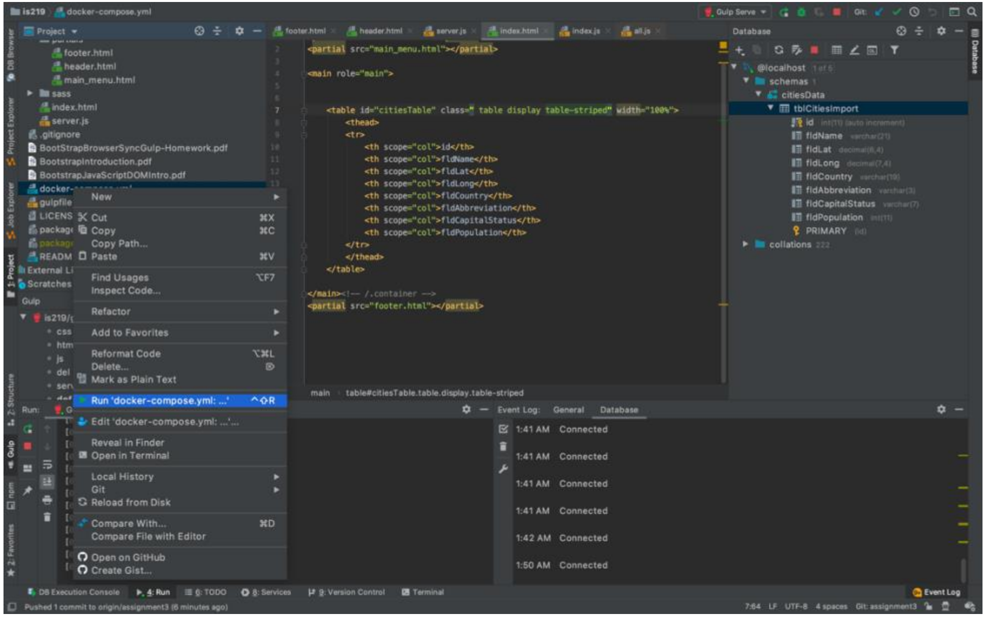
10 files committed: Task: Initial assignment 3 commit

1:50 AM Connected

Pushed 1 commit to origin/assignment3 (2 minutes ago)

Step 6 – Make sure that you run NPM install in the project directory

Step 7 – Right click on the “Docker-compose.yml” file in your project directory and click run



```
<partial src="main_menu.html"></partial>

<main role="main">


    <table id="citiesTable" class=" table display table-striped" width="100%">
        <thead>
        <tr>
            <th scope="col">id</th>
            <th scope="col">fldName</th>
            <th scope="col">fldLat</th>
            <th scope="col">fldLong</th>
            <th scope="col">fldCountry</th>
            <th scope="col">fldAbbreviation</th>
            <th scope="col">fldCapitalStatus</th>
            <th scope="col">fldPopulation</th>
        </tr>
        </thead>
    </table>

</main><!-- /.container -->
<partial src="footer.html"></partial>
```

Database

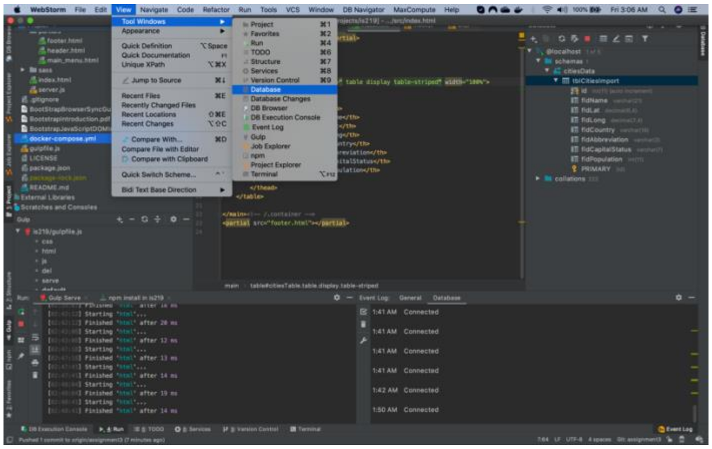
- @localhost 1 of 6
  - schemas 1
    - citiesData
      - tblCitiesImport
        - id int(11) (auto increment)
        - fldName varchar(21)
        - fldLat decimal(6,4)
        - fldLong decimal(7,4)
        - fldCountry varchar(19)
        - fldAbbreviation varchar(3)
        - fldCapitalStatus varchar(7)
        - fldPopulation int(11)
        - PRIMARY (id)
  - collations 222

Event Log:

```
1:41 AM  Connected
1:41 AM  Connected
1:41 AM  Connected
1:41 AM  Connected
1:42 AM  Connected
1:50 AM  Connected
```

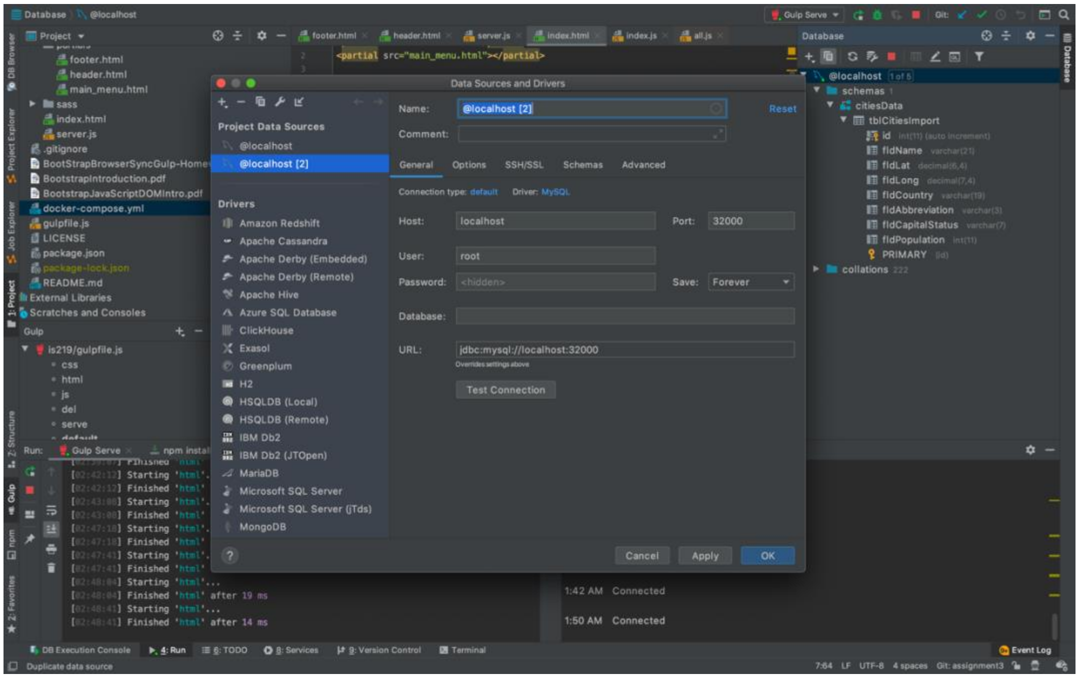
Pushed 1 commit to origin/assignment3 (6 minutes ago)

Step 8 – Click on “View” in Webstorm and go to “Tool Windows > Database” and see the panel on the right open.



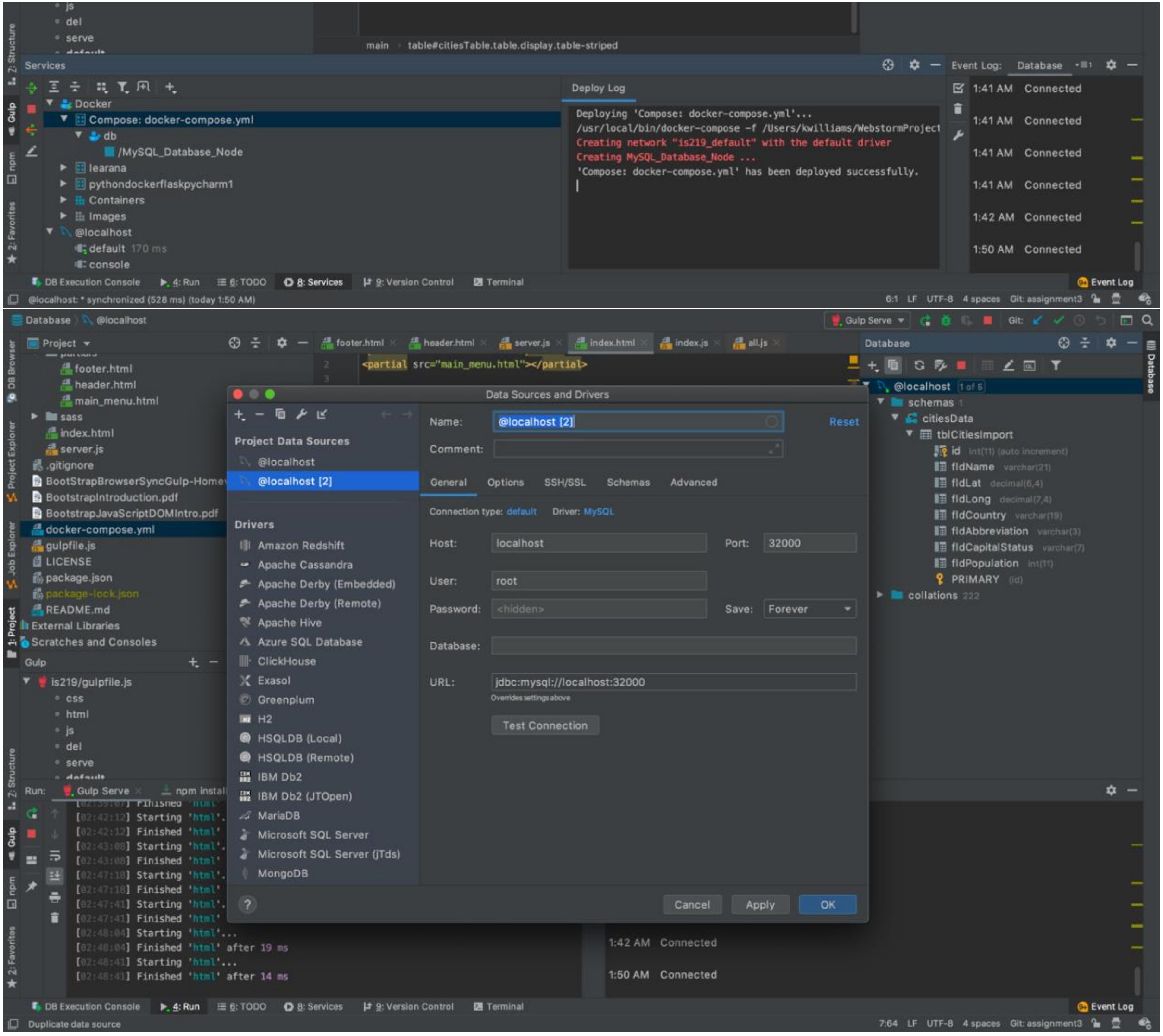
Step 9 – Setup the database connection to mysql, you need to be able to see the “citiesData” database and the “tblCitiesImport” table in the database view.  You need to click “download driver” at the bottom of the configuration screen, if you see a link for it.  You will not need to restart webstorm, just click the “download driver” link and it will just work.

Use the following settings:
Port: 32000
User: root
Password: root



Name: @localhost [2]
Comment:
Connection type: default    Driver: MySQL
Host: localhost    Port: 32000
User: root
Password: <hidden>    Save: Forever
Database:
URL: jdbc:mysql://localhost:32000
Overrides settings above

Step 10 - The database server needs to be running with docker and you can view docker by clicking “view” in Webstorm and going to the “Tool Windows > Services”.  When you do that you will see the screen shot below.  Yours will be slightly different, since I have various other systems running.  You should see “Compose: docker-compose.yml” and you can right click on it to put it “down” which destroys it and you can click on the “/MySQL_Database_node” to view the console output of the database server.  If you change your database schema you will need to put it down, so that when you bring it up with the run command that it will load the database initialization script.



Deploy Log

```
Deploying 'Compose: docker-compose.yml'...
/usr/local/bin/docker-compose -f /Users/kwilliams/WebstormProject
Creating network "is219_default" with the default driver
Creating MySQL_Database_Node ...
'Compose: docker-compose.yml' has been deployed successfully.
```

```
1:41 AM   Connected
1:41 AM   Connected
1:41 AM   Connected
1:41 AM   Connected
1:42 AM   Connected
1:50 AM   Connected
```

```
<partial src="main_menu.html"></partial>
```

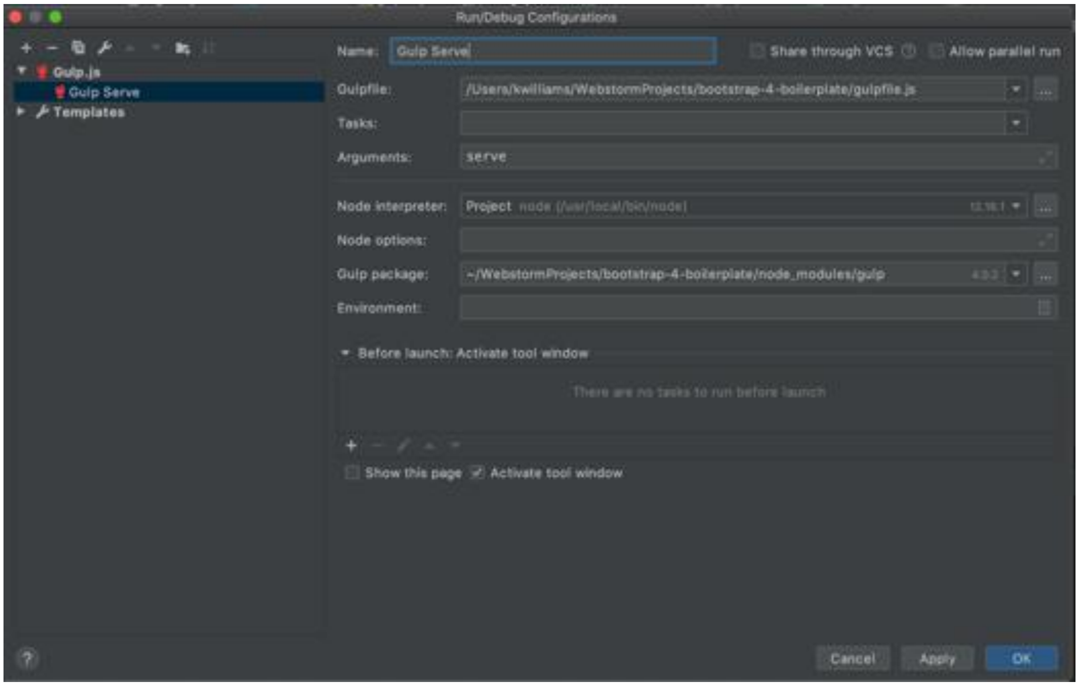
Name: @localhost [2]
Comment:
Connection type: default    Driver: MySQL
Host: localhost    Port: 32000
User: root
Password: <hidden>    Save: Forever
Database:
URL: jdbc:mysql://localhost:32000

```
@localhost 1 of 5
  schemas 1
    citiesData
      tblCitiesImport
        id  int(11) (auto increment)
        fldName  varchar(21)
        fldLat  decimal(6,4)
        fldLong  decimal(7,4)
        fldCountry  varchar(19)
        fldAbbreviation  varchar(3)
        fldCapitalStatus  varchar(7)
        fldPopulation  int(11)
        PRIMARY  (id)
  collations 222
```

```
[02:39:07] Finished 'html'
[02:42:12] Starting 'html'.
[02:42:12] Finished 'html'
[02:43:08] Starting 'html'.
[02:43:08] Finished 'html'
[02:47:18] Starting 'html'.
[02:47:18] Finished 'html'
[02:47:41] Starting 'html'.
[02:47:41] Finished 'html'
[02:48:04] Starting 'html'...
[02:48:04] Finished 'html' after 19 ms
[02:48:41] Starting 'html'...
[02:48:41] Finished 'html' after 14 ms
```

```
1:42 AM   Connected
1:50 AM   Connected
```

Step 11 – You need to make sure gulp is able to run the program.



Name: Gulp Serve

□ Share through VCS □ Allow parallel run

Gulpfile: /Users/kwilliams/WebstormProjects/bootstrap-4-boilerplate/gulpfile.js

Tasks:

Arguments: serve

Node interpreter: Project node (/usr/local/bin/node) 12.16.1

Node options:

Gulp package: ~/WebstormProjects/bootstrap-4-boilerplate/node_modules/gulp 4.0.2

Environment:

Before launch: Activate tool window

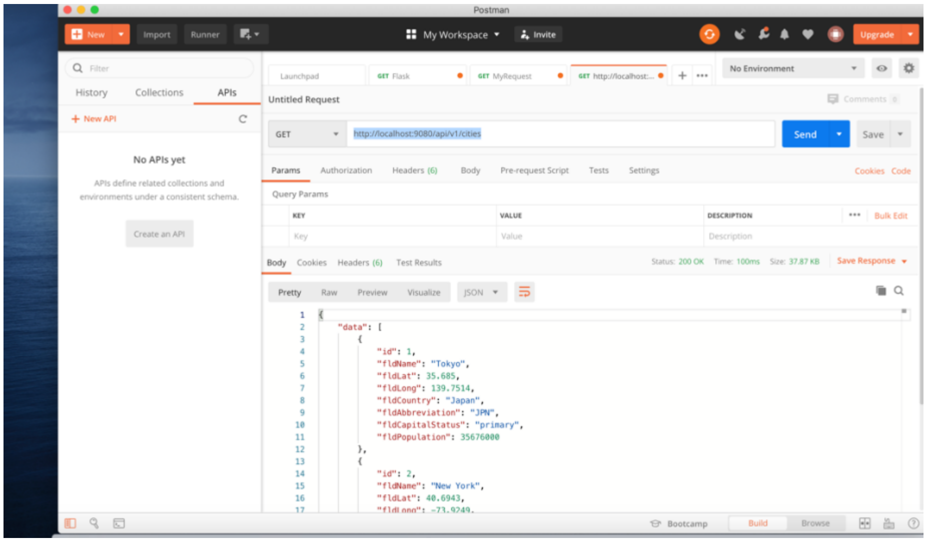
There are no tasks to run before launch

□ Show this page ☑ Activate tool window

Figure 1 - Put "serve" as the argument

Step 11 – Run the project and view it in the browser
Press the “Play” button in WebStorm to launch the site. This site no longer uses browser sync because the main focus of this is now loading data from the server and we are using “NodeMon” to monitor all our files and trigger building the system when the server.js file or other files are edited.

Step 12 – Once your project is running, you need to download and install Postman if you haven’t already. When you start postman you can put this URL in http://localhost:9080/api/v1/cities and send a get request to view the data returned by the endpoint.



Untitled Request

GET http://localhost:9080/api/v1/cities

| KEY | VALUE | DESCRIPTION |
|---|---|---|
| Key | Value | Description |

Status: 200 OK Time: 100ms Size: 37.87 KB

```
{
    "data": [
        {
            "id": 1,
            "fldName": "Tokyo",
            "fldLat": 35.685,
            "fldLong": 139.7514,
            "fldCountry": "Japan",
            "fldAbbreviation": "JPN",
            "fldCapitalStatus": "primary",
            "fldPopulation": 35676000
        },
        {
            "id": 2,
            "fldName": "New York",
            "fldLat": 40.6943,
            "fldLong": -73.9249,
```