

PROJECT SETUP

Before we start to write our predefined operations, we should start with some sort of uniform folder structure so that following along with these articles is as clear as possible.

Part 1: Folder Structure

Step 1: Create a folder in the backend along side the server.js file and name it “routes.”

Step 2: Inside this folder we will code all of our predefined operations related to each collection type in our database. Start by creating a file named cities.js.

Step 3: Navigate back to the server.js file and make a call to “app.use” and pass in two arguments, the api route and the relative path to the file where we will defining our route methods.

```
app.use('/cities', require('./routes/cities'))
```

Step 4: Back in our cities.js file let’s write a test route just to make sure everything is hooked up correctly. At the top of our file require “express” by setting it to a variable and create an instance of the express router by setting it to a variable called “router.”

```
const express = require('express')
const router = express.Router()
```

Step 5: Write a test route.

```
router.get('/test', (req, res) => {
// We pass two arguments into our get method. The path and a
call back function with our request object and response object
as parameters.
// Since we made the call to app.use() in our server.js file we
only need to pass in “/test” as our first argument here, however
this refers to the endpoint “localhost:3000/cities/test”.
  res.send({ msg: 'Cities test route.' })
})
// In our response object we send back a simple string in JSON
format.
```

Part 2: Testing on Postman

Step 1: Make sure our server is running and move into the Postman desktop application.

Step 2: Make a “GET” request to “localhost:3000/cities/test” and hit send.

Step 3: You should receive the message “Cities test route” in your response body. See Figure 1 for reference.

Figure 1:

