

DELETE

The last route we will be writing for the CityCreator project will handle deleting resources. We will again be targeting specific resources by their id. Luckily this route is fairly simple.

Part 1: Delete Route Method

Step 1: Begin writing our route method

```
router.delete('/:id', (req, res) => {  
  // For deleting resources we use the HTTP delete method and pass  
  // as our first argument our endpoint with the dynamic parameter  
  // for our document ids.
```

Step 2: Finding and deleting

```
    City.findByIdAndDelete(req.params.id)  
  // The mongoose method findByIdAndDelete does exactly what it  
  // says. We pass in as the only argument the object id which we get  
  // from the dynamic “:id” parameter of our endpoint.
```

Step 3: Checking for conditional logic

```
    .then(city => {  
      if (!city) {  
        return res.status(404).send('City not found.')  
      }  
    })  
  // Here we again make sure the city we wish to remove from our  
  // database actually exists. If not we immediately return from our  
  // function and send our custom error message as the response.
```

Step 4: Sending back a success message.

```
    res.send({ msg: 'City deleted.' })  
  })  
  .catch(err => res.status(500).send(err.message))  
})  
  // Rather than sending back the document that we removed, it is  
  // sometimes preferred to send a success message. Here if  
  // everything goes well we will receive 'City deleted.' in our  
  // response.
```

Part 2: Testing on Postman

Step 1: Create a duplicate City with a Post request to 'localhost:500/cities/'

Step 2: Copy the object id of the duplicate city to your clipboard.

Step 3: Set the HTTP method to a “DELETE” request and set the end point to “localhost:5000/api/cities/”

Step 4: Paste the City’s object to the end of your route.

Step 5: Fire off the request. If successful you should receive the “City deleted.” message.

Step 6: Fire off the same request again, this time you should receive the “City not found.” message, as now the city should no longer exist in the database.

Step 7: If unsuccessful use the error message to debug.

- See Figure 1 for reference.

Figure 1:

