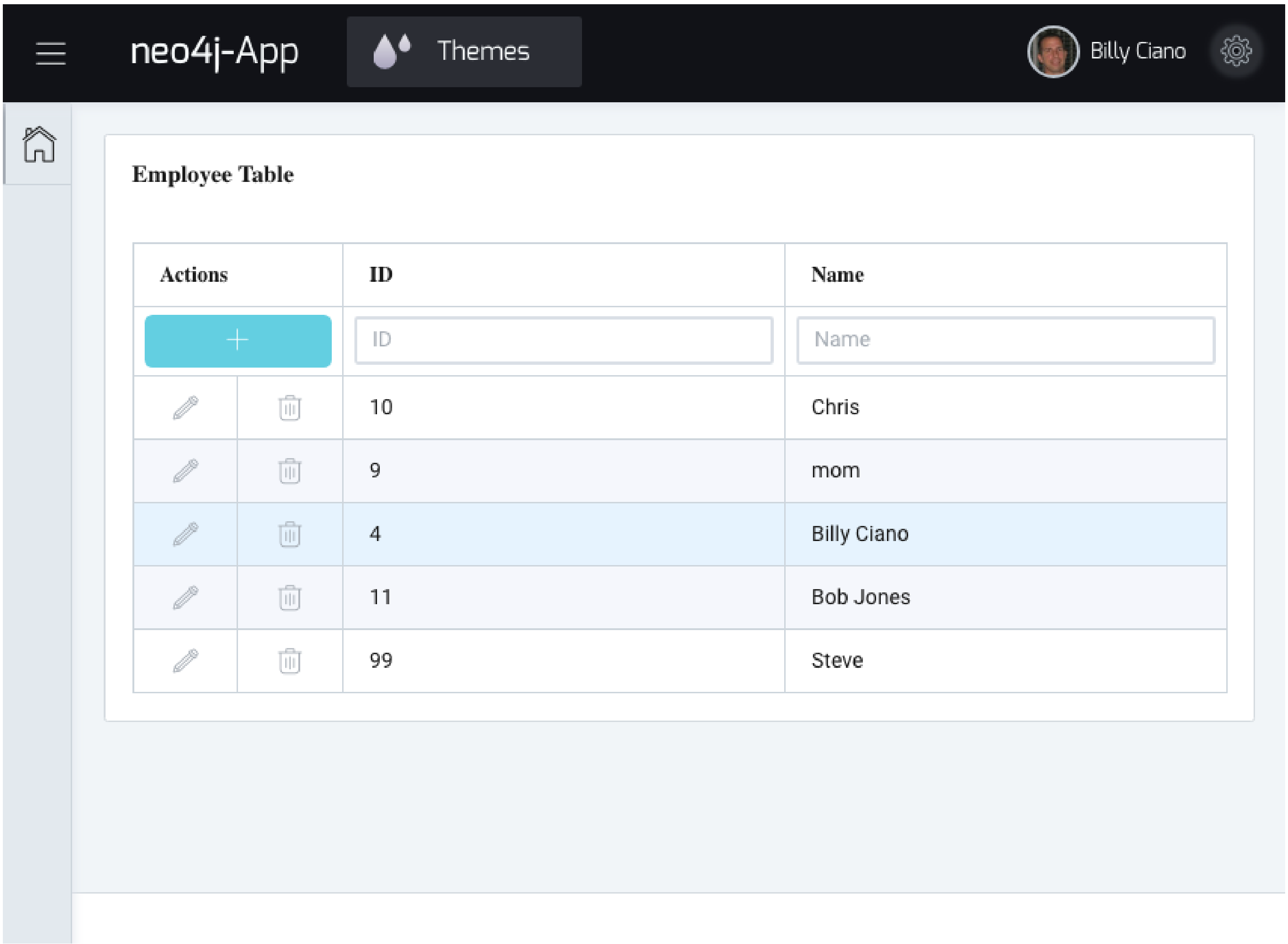
Neo-App

Intro

Simple web application which can be used to drive the API. THe application was built using Angular CLI (7.1.2), and leverages a few Angular web frameworks, for asthetics, including ngx-admin, and nebular. The GIT project consists of a base ngx-admin application, with additions for an Employee table. The app was built in "production-mode", and deployed to a web server (Tomcat) on the EC2 instance. A dashboard web component exists which contains a single table with actions that communicate with the backend services from the Angular Employee service to the Express REST server via port 4114.

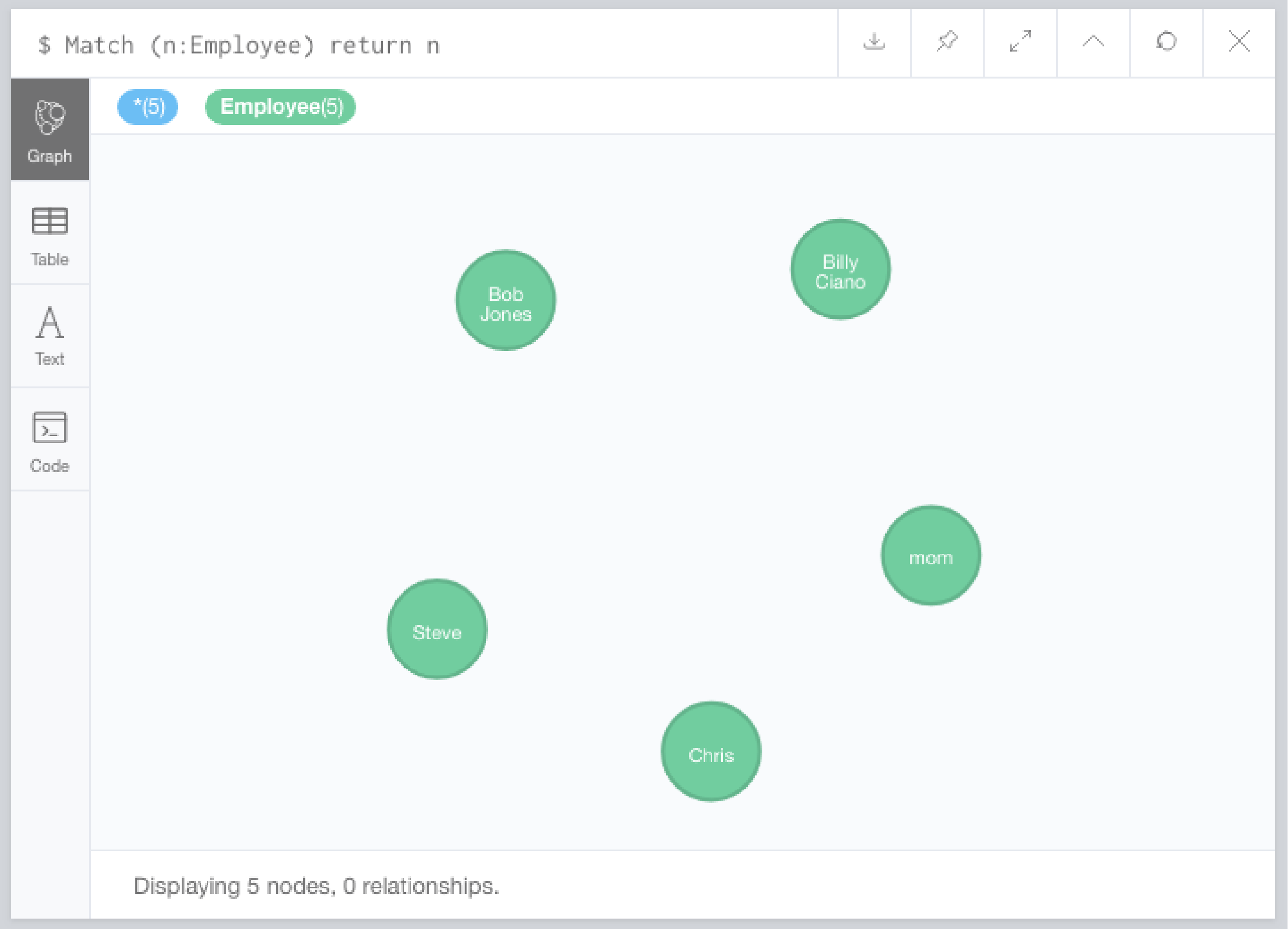
Application URL: http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:8080/

Neo4j Browser: http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:7474/browser/



Employee Table

Actions		ID	Name
+		ID	Name
✎	🗑	10	Chris
✎	🗑	9	mom
✎	🗑	4	Billy Ciano
✎	🗑	11	Bob Jones
✎	🗑	99	Steve



```
$ Match (n:Employee) return n
```

Displaying 5 nodes, 0 relationships.

UI Setup

If you would like to setup the Express server locally, follow these steps:

1. Have the latest stable versions of node/npm and git installed in your dev environment
2. clone the git repo for this project
3. `node install`
4. `ng start`

Express Server

If you would like to setup the Express server locally, follow these steps:

1. Have the latest stable versions of node/npm and git installed in your dev environment
2. clone the git repo for this project
3. `node install`
4. `cd server`
5. `node index.js` THe server will startup on port 4114. You can test that it is running by following this URL: http://localhost:4114/api/employee

Neo-App REST API

The Middle Tier server is implemented using NodeJS/Express, and the Neo4J Bolt Driver to access the database. REST implementations for the EMployee resources were created for GET, POST, PUT, and DELETE.

If you are familiar with PostMan, I have published the API documentation, which can be accessed on their public repository at: https://documenter.getpostman.com/view/6011757/RzfiJUYp (see screenshots of PostMan at the bottom of this doc)

Basic CURL commands to exercise the API

- GET All Employees:

```
curl --location --request GET "http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/"
```

- Create Employee:

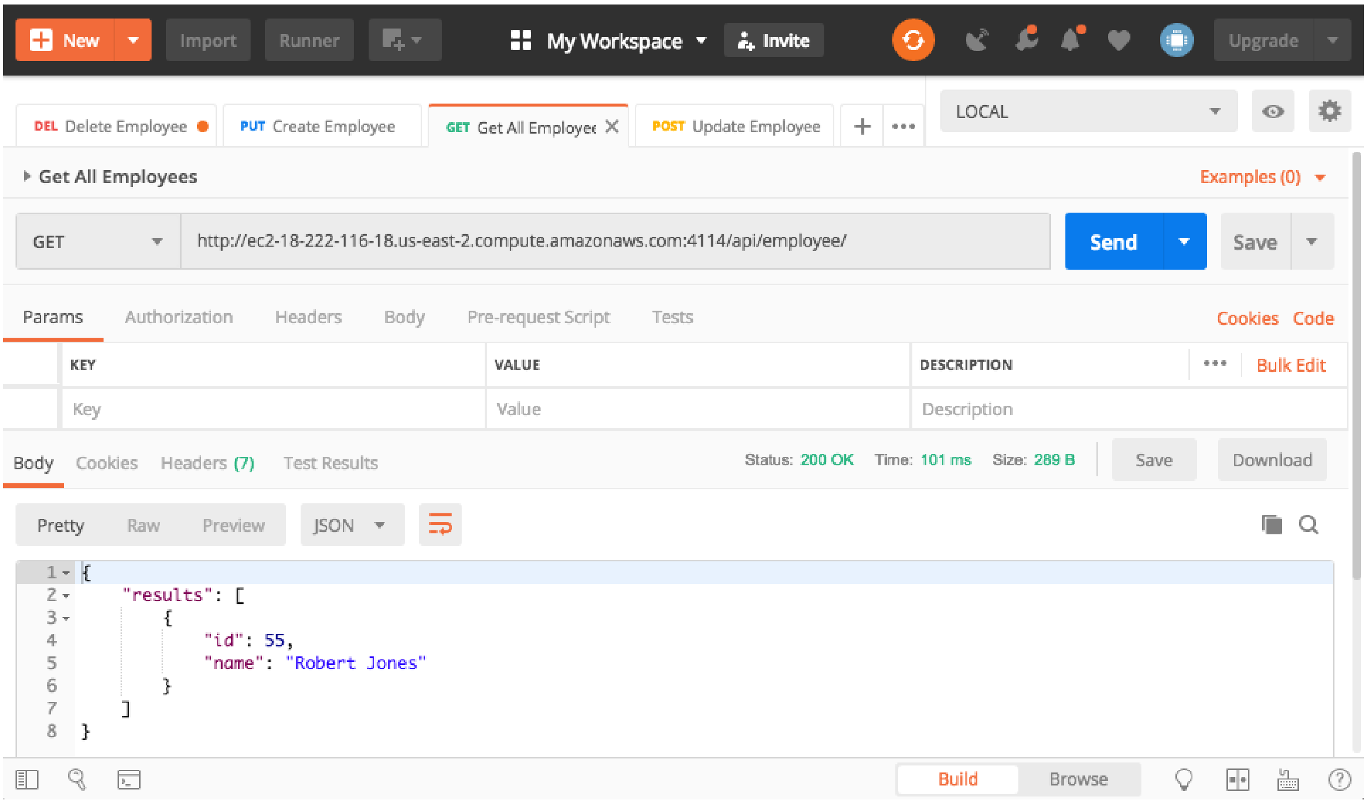
```
curl --location --request PUT "http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/" \
--header "Content-Type: application/x-www-form-urlencoded" \
--data "id=65&name=bob%20smith"
```

- Remove Employee:

```
curl --location --request DELETE "http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/65" \
  --data ""
```

- Update Employee:

```
curl --location --request POST "http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/65" \
  --header "Content-Type: application/x-www-form-urlencoded" \
  --data "id=55&name=Robert%20Jones"
```



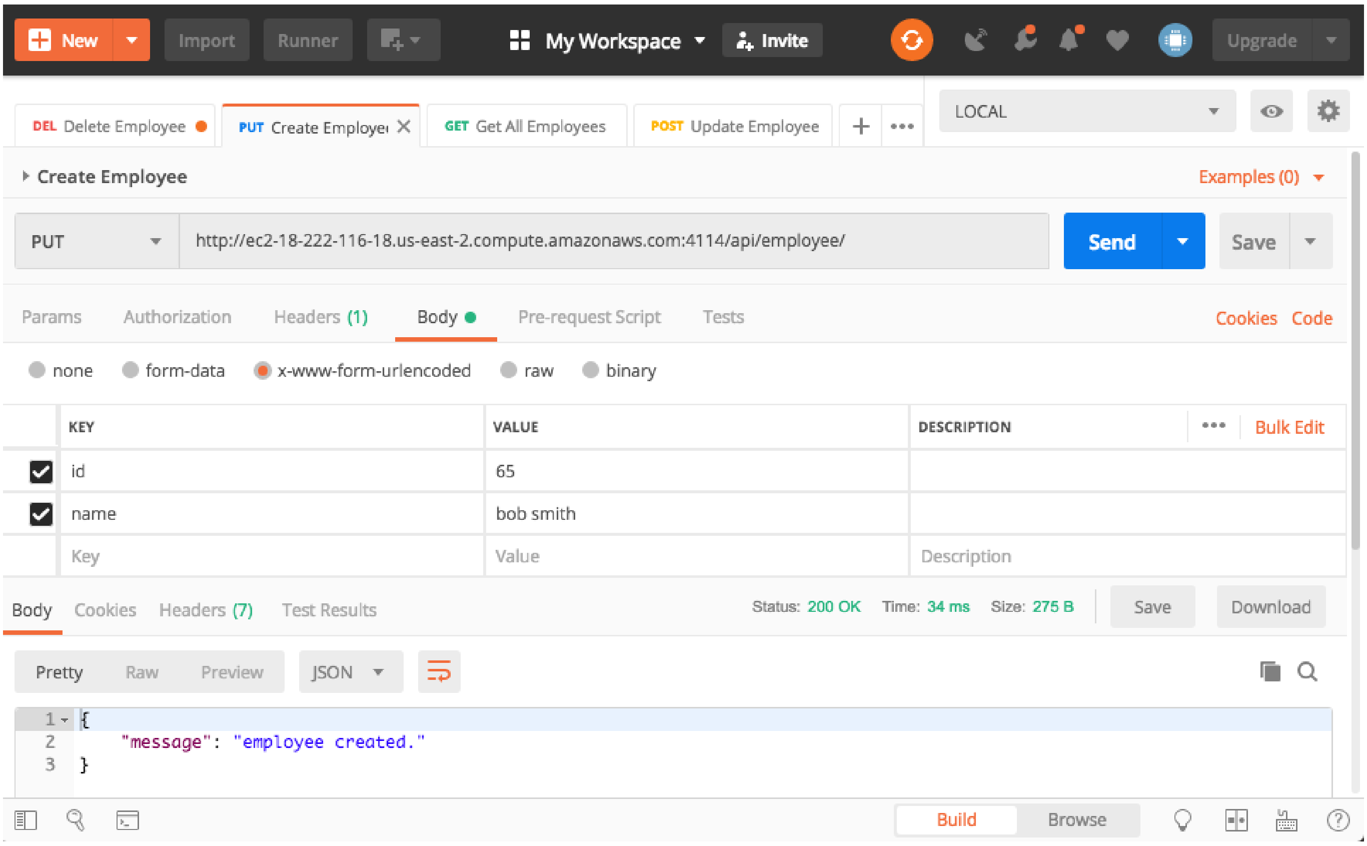
Get All Employees

GET http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/

KEY	VALUE	DESCRIPTION
Key	Value	Description

Status: 200 OK Time: 101 ms Size: 289 B

```
{
    "results": [
        {
            "id": 55,
            "name": "Robert Jones"
        }
    ]
}
```



Create Employee

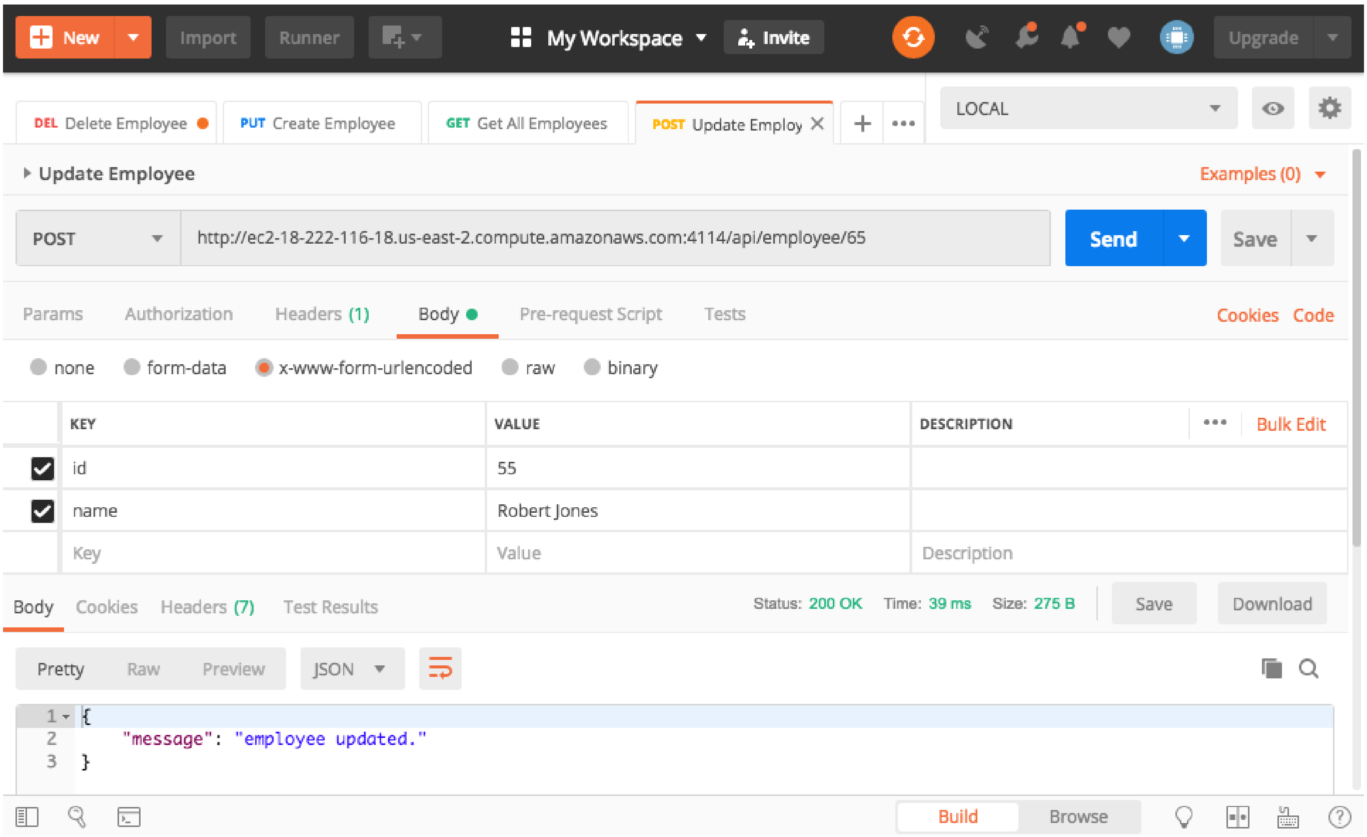
PUT http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	id	65	
☑	name	bob smith	
	Key	Value	Description

Status: 200 OK Time: 34 ms Size: 275 B

```
{
    "message": "employee created."
}
```



Update Employee

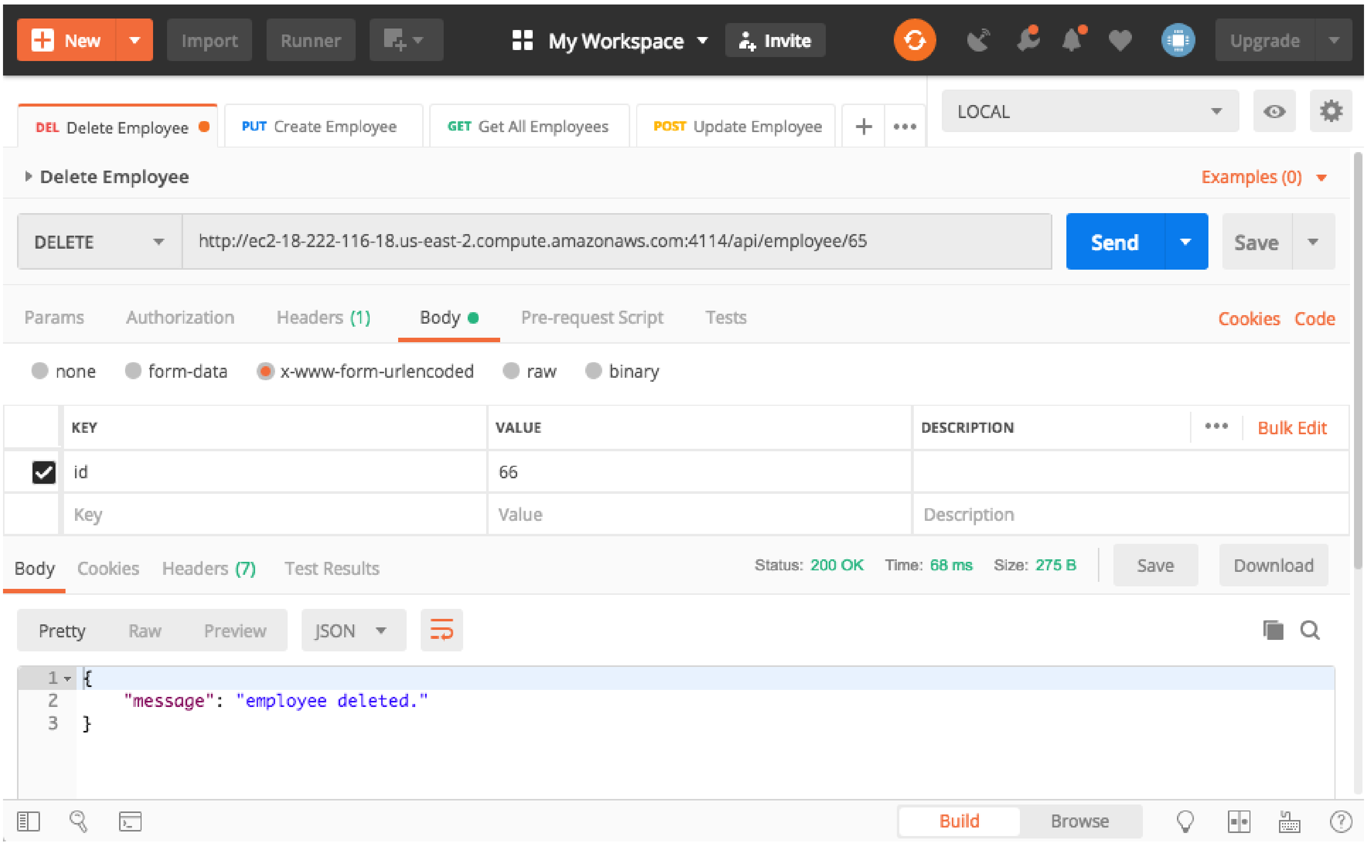
POST http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/65

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	id	55	
☑	name	Robert Jones	
	Key	Value	Description

Status: 200 OK Time: 39 ms Size: 275 B

```
{
    "message": "employee updated."
}
```



Delete Employee

DELETE http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:4114/api/employee/65

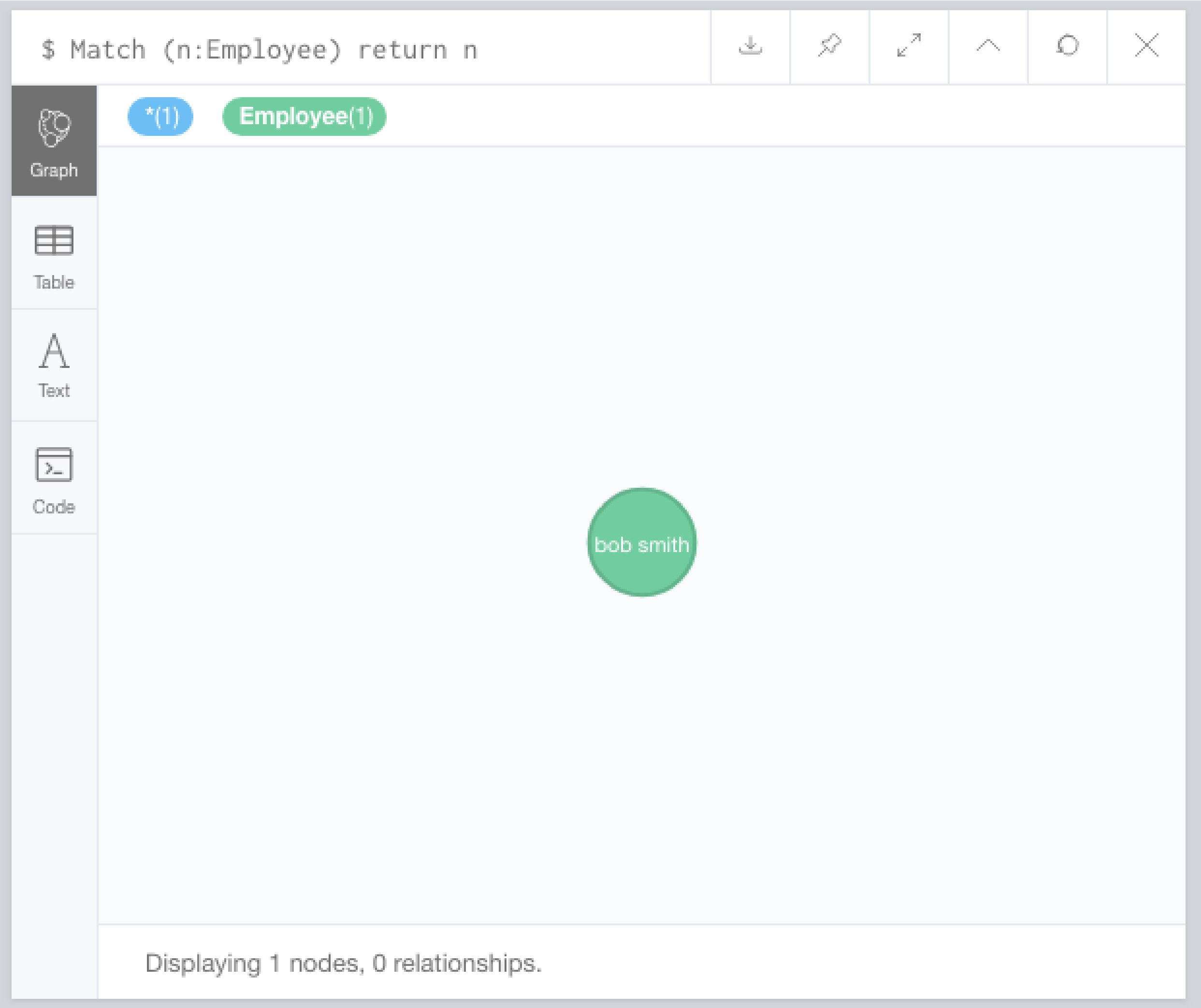
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary

	KEY	VALUE	DESCRIPTION
☑	id	66	
	Key	Value	Description

Status: 200 OK Time: 68 ms Size: 275 B

```
{
    "message": "employee deleted."
}
```

Neo4j Browser: http://ec2-18-222-116-18.us-east-2.compute.amazonaws.com:7474/browser/



```
$ Match (n:Employee) return n
```

Displaying 1 nodes, 0 relationships.

Source Code - Employee Files

- neo-app/server/index.js (Express REST API)
- neo-app/src/app/pages/dashboard/dashboard.component.ts
- neo-app/src/app/pages/dashboard/dashboard.component.html
- neo-app/src/app/@core/data/employee.service.ts