ACCOLITE UNIVERSITY, JULY 2020

REST Web Services in Java

Assignment Submission

By: Ayush Malik

I have made the REST API using Node.js. It uses 2 dependencies: 'express' and 'body-parser'. Please run 'npm install' after cloning the repository to resolve all dependencies.

Structure of a To-Do Object

```
{
	"id": 5,
	"value": "This is a sample To-Do"
}
```

Server Address

```
http://127.0.0.1:3000
```

API Reference

The following table summarizes the operations supported by my API.

URI	HTTP Method	Parameter	POST Body	Functionality Performed
listAll	GET	-	-	Display the complete To-Do list.
-	GET	id	-	Display the To-Do whose 'id' is equal to the parameter in the request.
add	POST	-	JSON	Add the new To-Do given in the JSON body to the To-Do list.
update	POST	-	JSON	Updating an existing To-Do whose 'id' in the JSON body matches an existing

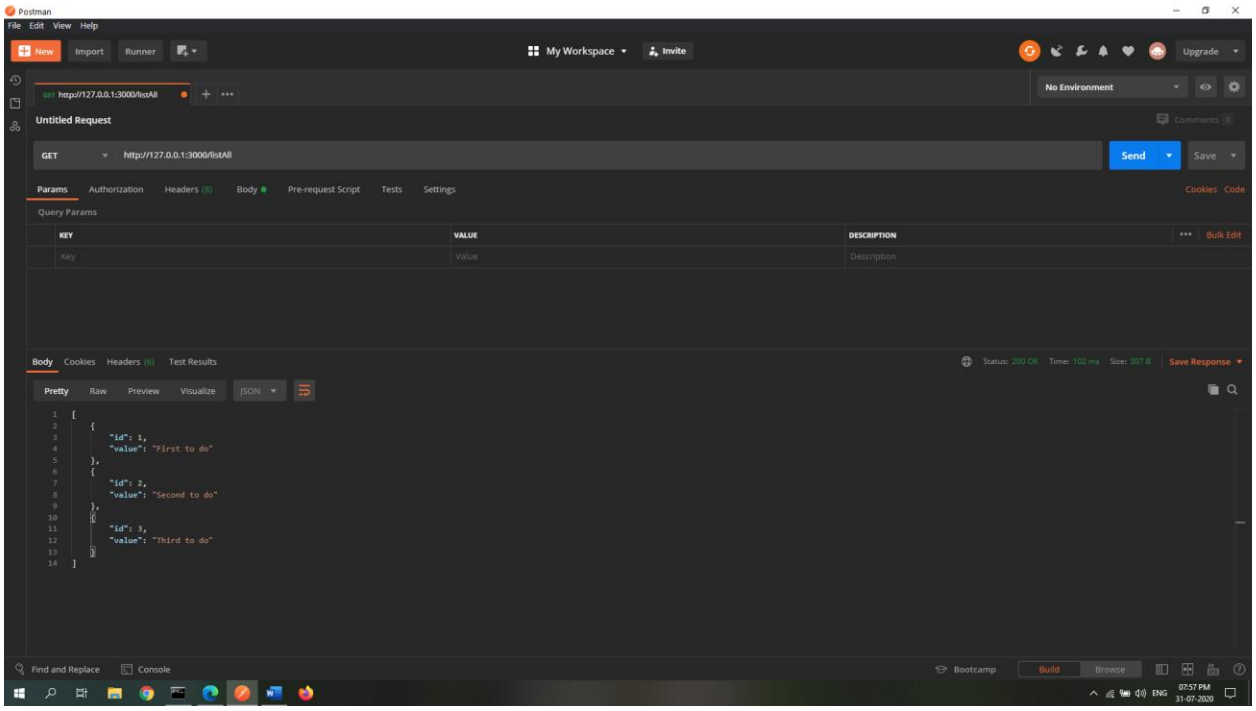
				To-Do's 'id' in the To-Do list.
delete	DELETE	id	-	Delete the To-do whose 'id' is equal to the parameter in the request.

Use of each individual operation with sample commands and outputs has been explained in depth below.

GET: Display the complete To-Do list

This request returns the complete To-Do list.

GET: `http://127.0.0.1:3000/listAll`



GET http://127.0.0.1:3000/listAll

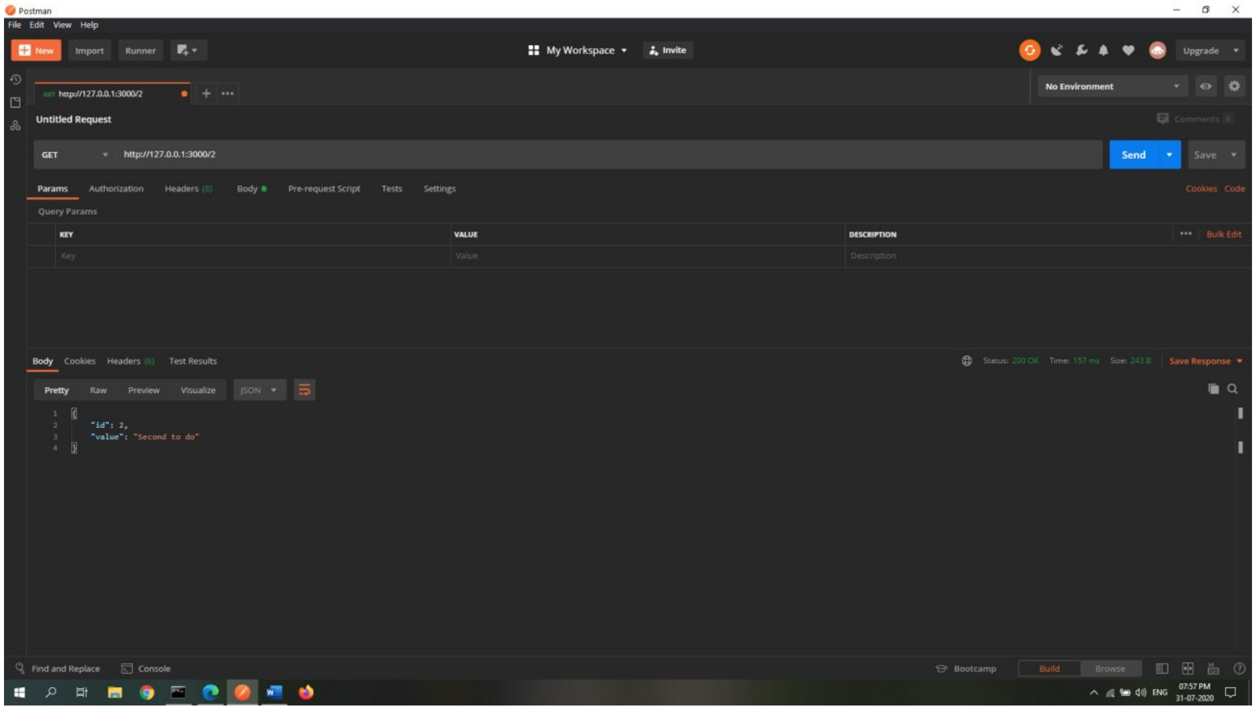
Status: 200 OK Time: 102 ms Size: 307 B

```
[
    {
        "id": 1,
        "value": "First to do"
    },
    {
        "id": 2,
        "value": "Second to do"
    },
    {
        "id": 3,
        "value": "Third to do"
    }
]
```

GET: Displaying To-Do with a particular ‘id’

This request returns a To-Do whose ‘id’ is equal to the parameter in the request. If no match is found, then a message stating “No Element with the given id found” is returned.

GET: `http://127.0.0.1:3000/2`



GET http://127.0.0.1:3000/2

Status: 200 OK Time: 157 ms Size: 243 B

```
{
    "id": 2,
    "value": "Second to do"
}
```

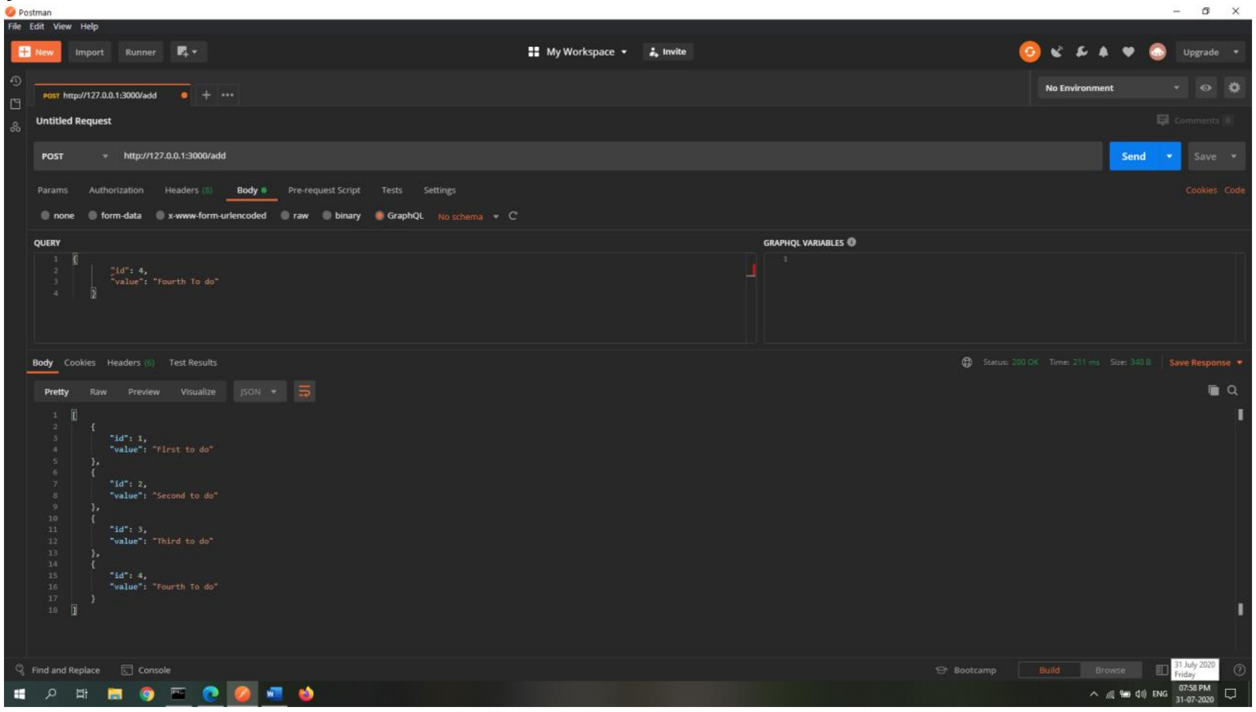
POST: Adding a new To-Do

This request adds a new To-Do given in the request's JSON body to the To-Do list and returns the new To-Do-List.

POST: `http://127.0.0.1:3000/add`

Body:

```
{
        "id": 4,
        "value": "Fourth To do"
}
```



Untitled Request

POST http://127.0.0.1:3000/add

Body: GraphQL

QUERY

```
{
    "id": 4,
    "value": "Fourth To do"
}
```

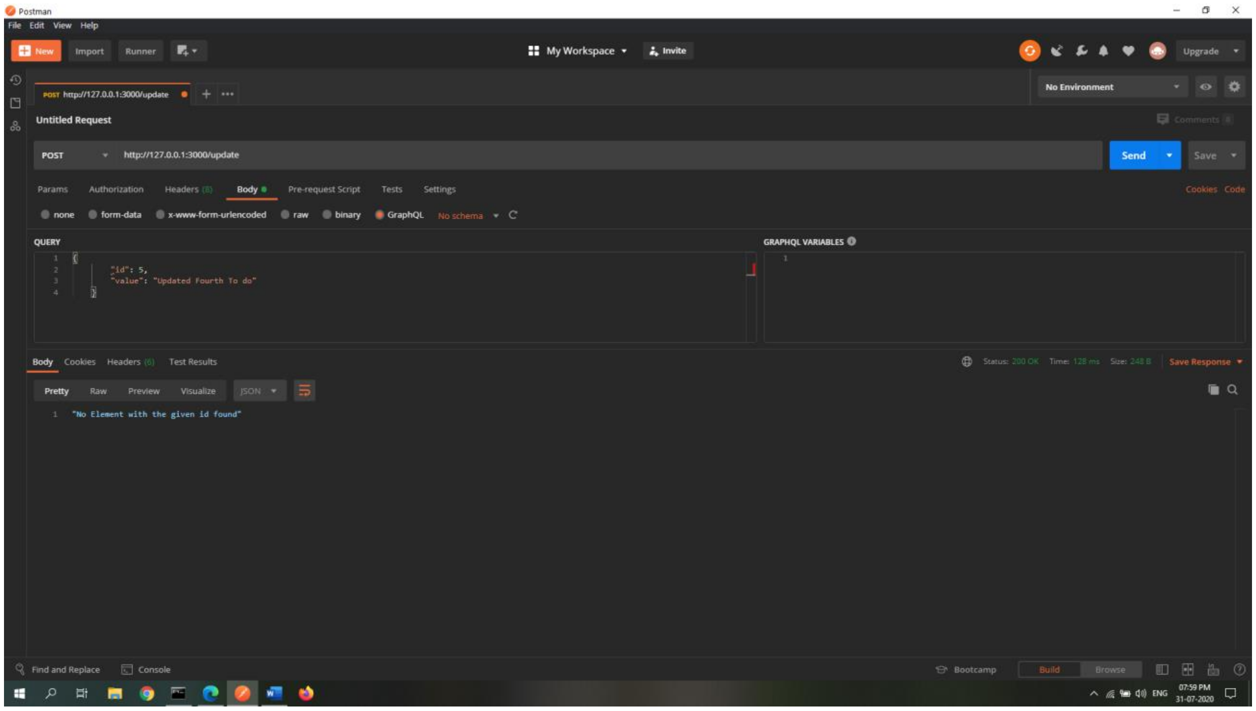
Status: 200 OK Time: 211 ms Size: 340 B

```
[
    {
        "id": 1,
        "value": "First to do"
    },
    {
        "id": 2,
        "value": "Second to do"
    },
    {
        "id": 3,
        "value": "Third to do"
    },
    {
        "id": 4,
        "value": "Fourth To do"
    }
]
```

POST: Updating an existing To-Do

This request updates an existing To-Do whose ‘id’ in the JSON body matches an existing To-Do’s ‘id’ in the To-Do list and returns the updated To-Do. If no match is found, then a message stating “No Element with the given id found” is returned.

POST: `http://127.0.0.1:3000/update`
Body:
```
{
        "id": 5,
        "value": "Updated Fourth To do"
}
```



POST http://127.0.0.1:3000/update

Untitled Request

POST http://127.0.0.1:3000/update

Body: GraphQL

QUERY
```
{
    "id": 5,
    "value": "Updated Fourth To do"
}
```

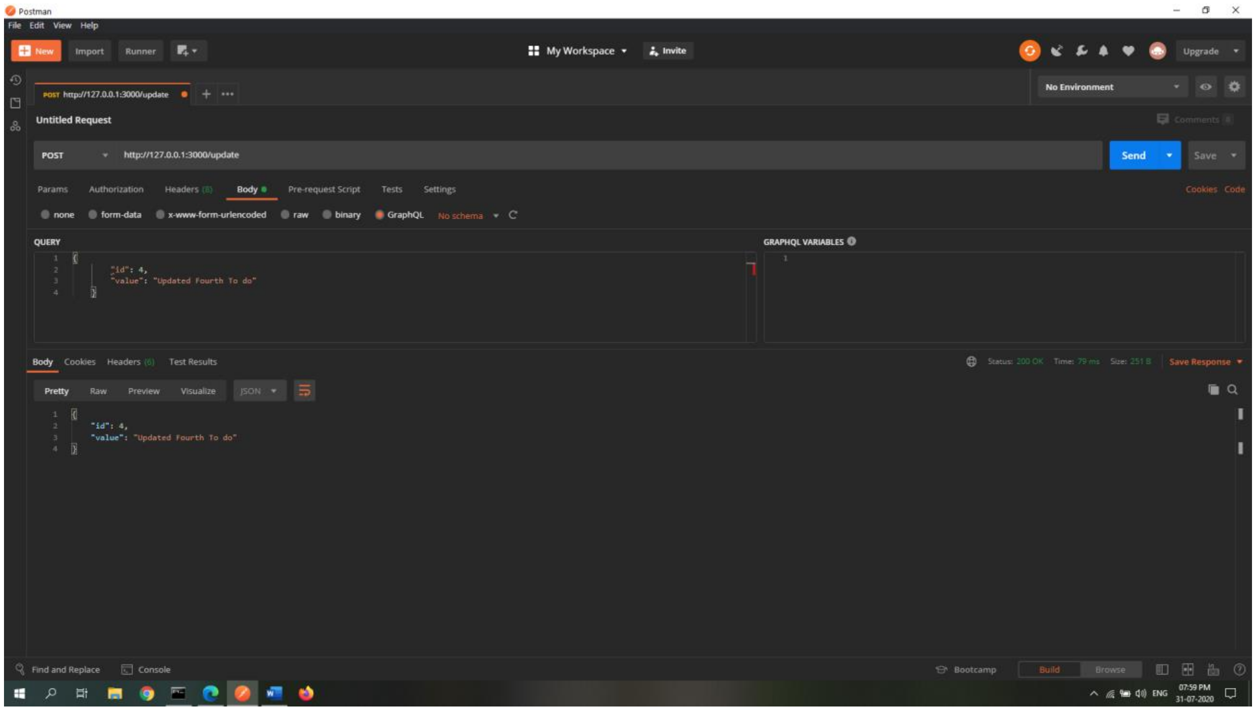
Status: 200 OK Time: 128 ms Size: 248 B

```
"No Element with the given id found"
```

POST: `http://127.0.0.1:3000/update`

Body:

```
{
        "id": 4,
        "value": "Updated Fourth To do"
}
```



Untitled Request

POST http://127.0.0.1:3000/update

Body: GraphQL

QUERY

```
{
    "id": 4,
    "value": "Updated Fourth To do"
}
```

Status: 200 OK Time: 79 ms Size: 251 B

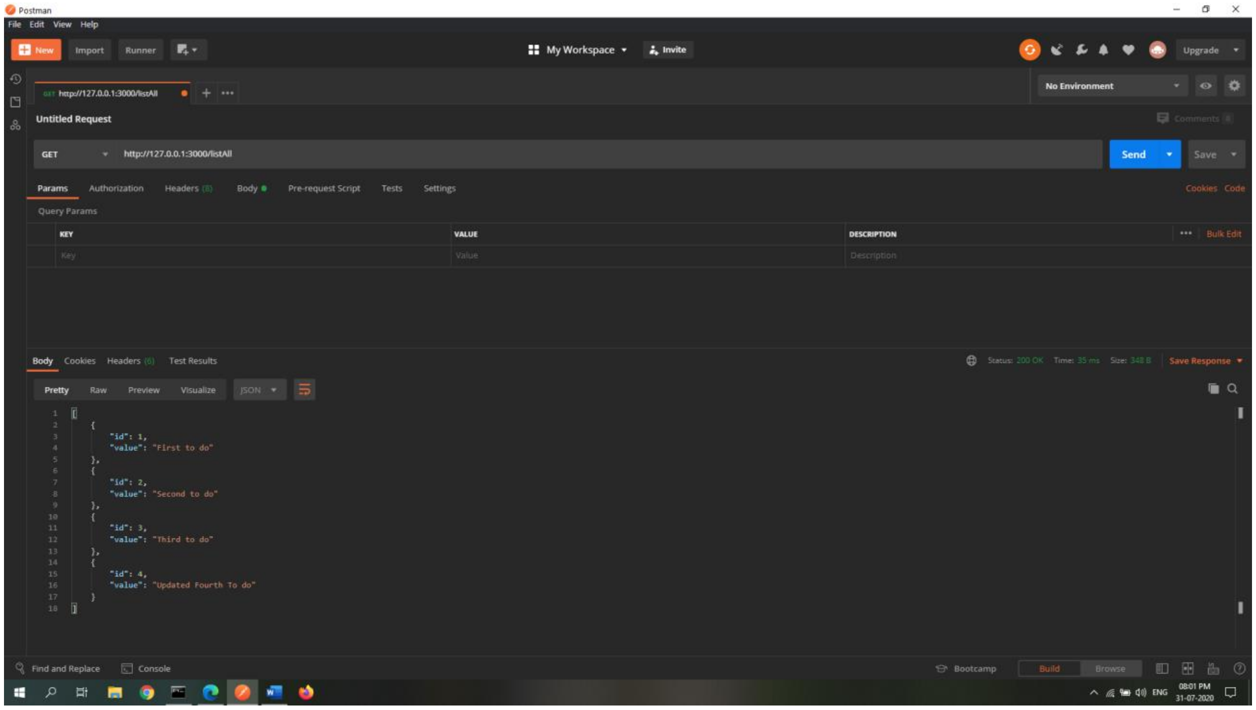
```
{
    "id": 4,
    "value": "Updated Fourth To do"
}
```

07:59 PM 31-07-2020

DELETE: Deleting an existing To-Do

This request deletes the To-do whose 'id' is equal to the parameter in the request. It returns the updated To-Do list after deletion of the record. If no match is found, then a message stating "No Element with the given id found" is returned.

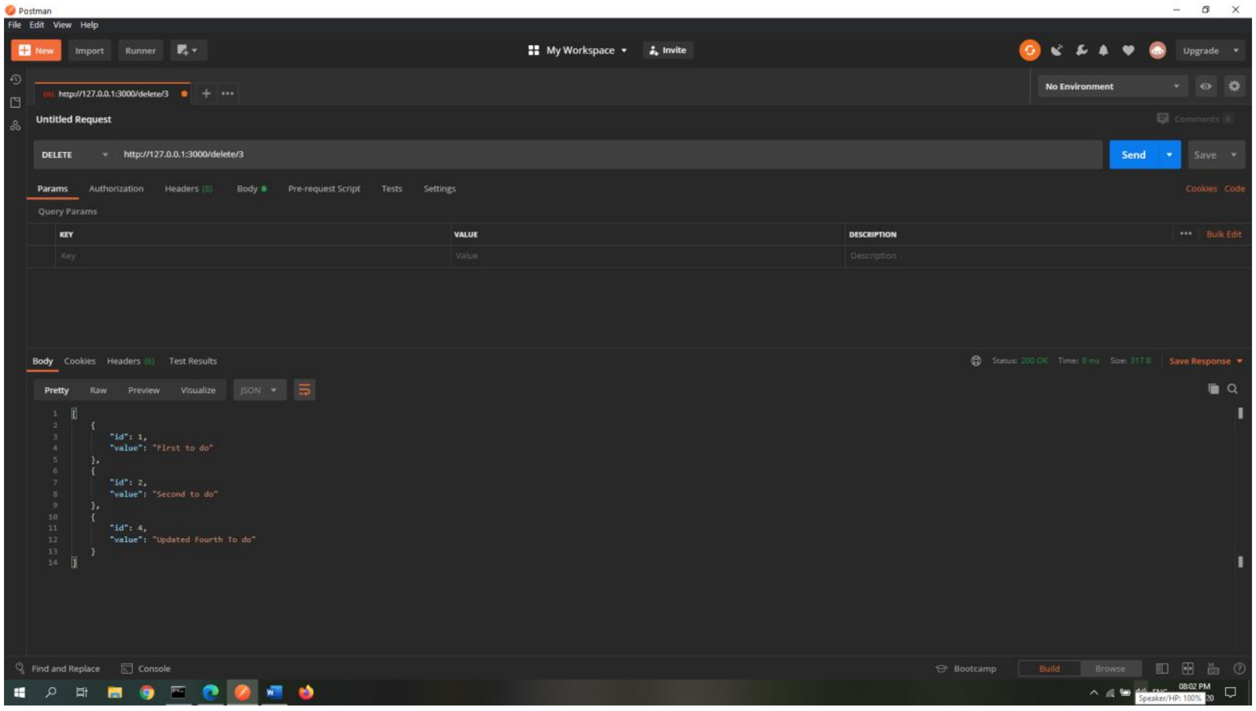
GET: `http://127.0.0.1:3000/listAll`



```
GET    http://127.0.0.1:3000/listAll
Status: 200 OK   Time: 35 ms   Size: 348 B
```

```
[
    {
        "id": 1,
        "value": "First to do"
    },
    {
        "id": 2,
        "value": "Second to do"
    },
    {
        "id": 3,
        "value": "Third to do"
    },
    {
        "id": 4,
        "value": "Updated Fourth To do"
    }
]
```

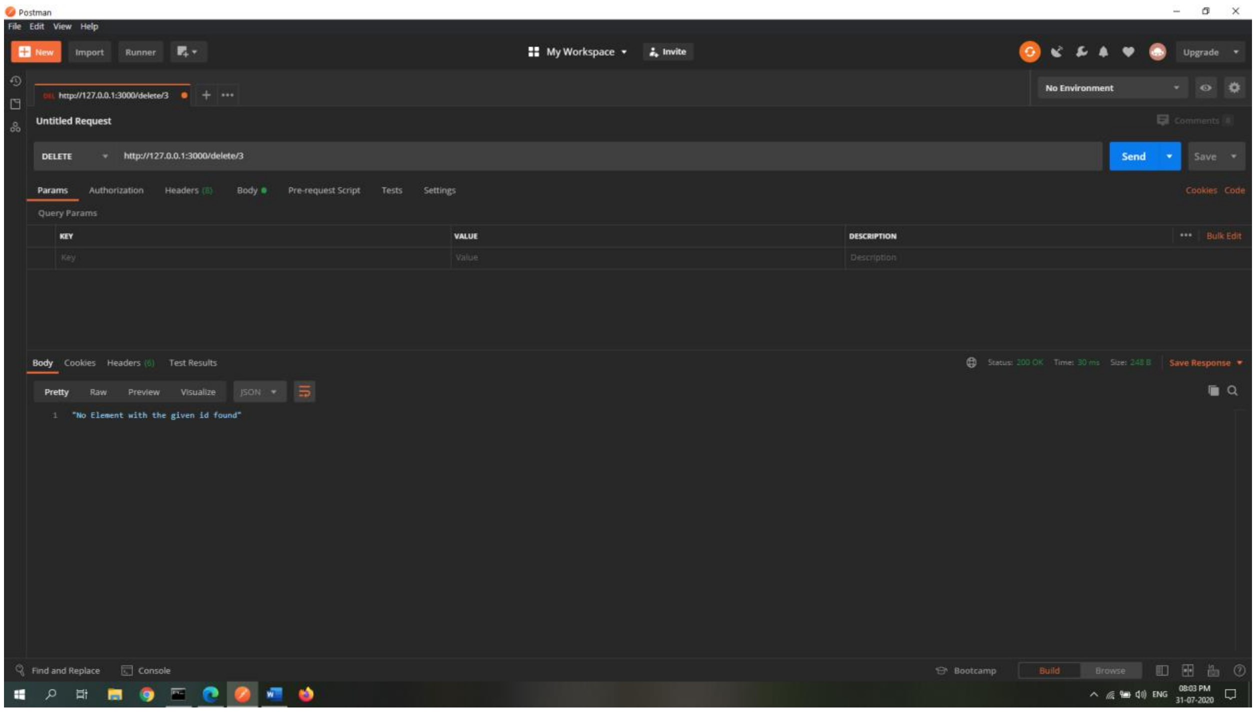
DELETE: `http://127.0.0.1:3000/delete/3`



```
DELETE    http://127.0.0.1:3000/delete/3
Status: 200 OK   Time: 8 ms   Size: 317 B
```

```
[
    {
        "id": 1,
        "value": "First to do"
    },
    {
        "id": 2,
        "value": "Second to do"
    },
    {
        "id": 4,
        "value": "Updated Fourth To do"
    }
]
```

DELETE: `http://127.0.0.1:3000/delete/3`


Untitled Request
DELETE
http://127.0.0.1:3000/delete/3
Send
Status: 200 OK Time: 30 ms Size: 248 B

```
"No Element with the given id found"
```