

Name: Dominic Gopalakrishnan

ID: 101289239

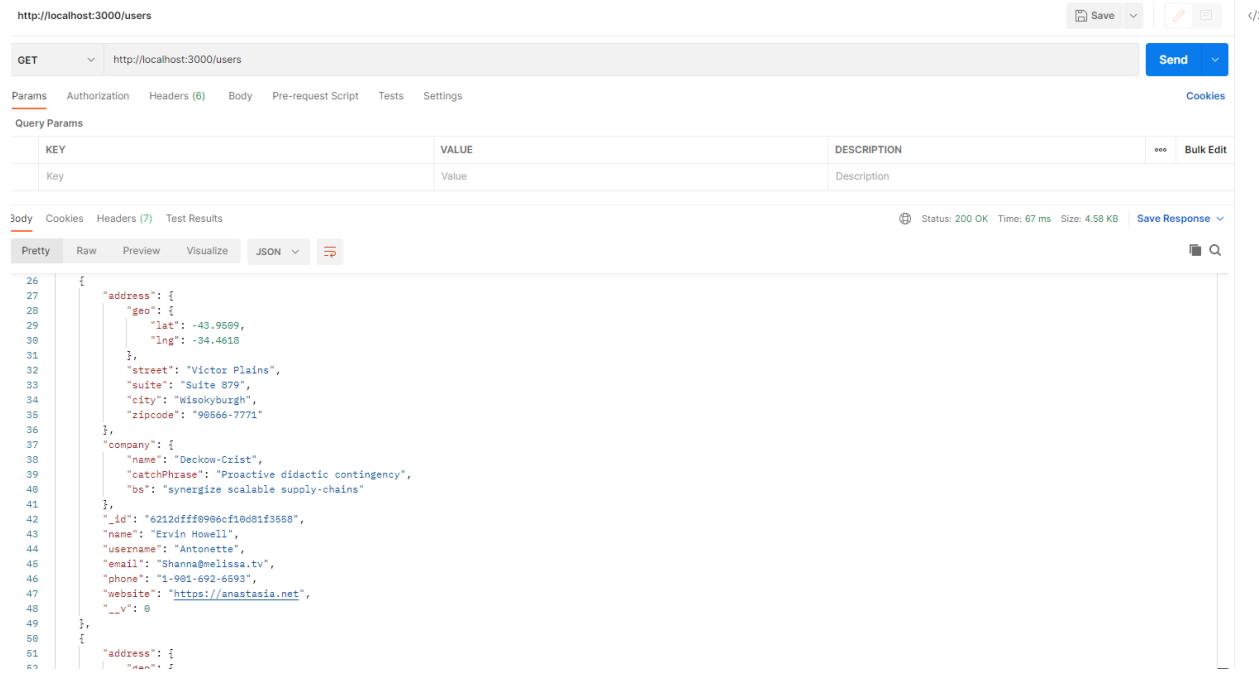
CRN: 49746

Full Stack Development

Lab 4 – Mongoose Validation

Screenshots:

Postman get:



The screenshot shows a Postman interface with the following details:

- Method: GET
- URL: <http://localhost:3000/users>
- Headers: None
- Body: None
- Query Params:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		
- Tests: None
- Settings: None
- Cookies: None
- Body tab selected
- Raw tab selected
- Preview tab selected
- Visualize tab selected
- JSON tab selected
- Status bar: Status: 200 OK Time: 67 ms Size: 4.58 KB Save Response

The response body is a JSON object representing a user document:

```
26  {
27   "address": {
28     "geo": [
29       {
30         "lat": -43.95009,
31         "lng": -34.4616
32       },
33       {
34         "street": "Victor Plains",
35         "suite": "Suite 879",
36         "city": "Wisokyburgh",
37         "zipcode": "90866-7771"
38       },
39       {
40         "company": {
41           "name": "Deckow-Crist",
42           "catchPhrase": "Proactive didactic contingency",
43           "bs": "synergize scalable supply-chains"
44         },
45         "_id": "6212dfffb9906cf10d81f3556",
46         "name": "Ervin Howell",
47         "username": "Antonette",
48         "email": "Shanna@melissa.tv",
49         "phone": "1-901-692-6593",
50         "website": "https://anastasia.net",
51         "__v": 0
52       },
53       {
54         "address": [
55           {
56             "geo": [
57               {
58                 "lat": -43.95009,
59                 "lng": -34.4616
60               }
61             ]
62           }
63         ]
64       }
65     ]
66   }
67 }
```

Error filled postman body:

ooo http://localhost:3000/users

POST http://localhost:3000/users

Params Authorization Headers (8) **Body** ● Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```
1  {
2   ... "name": "Tester-1",
3   ... "username": "Bre",
4   ... "email": "Sincereapril.biz",
5   ... "address": {
6     ... "street": "Kulas Light",
7     ... "suite": "Apt. 556",
8     ... "city": "Gwenborough",
9     ... "zipcode": "9299811111-3874",
10    ... "geo": {
11      ... "lat": "-37.3159",
12      ... "lng": "81.1496"
13    },
14  },
15  ... "phone": "1-712312370-736-8031",
16  ... "website": "test",
17  ... "company": {
18    ... "name": "Romaguera-Crona",
19    ... "catchPhrase": "Multi-layered client-server neural-net",
20    ... "bs": "harness real-time e-markets"
21  },
22 }
```

Results/error validation:

```

1
2   "errors": {
3     "address.zipcode": {
4       "name": "ValidatorError",
5       "message": "Zipcode must be in valid format",
6       "properties": {
7         "message": "Zipcode must be in valid format",
8         "type": "user defined",
9         "path": "address.zipcode",
10        "value": "9299811111-3874"
11      },
12      "kind": "user defined",
13      "path": "address.zipcode",
14      "value": "9299811111-3874"
15    },
16    "address.city": {
17      "name": "ValidatorError",
18      "message": "City must be in valid format!",
19      "properties": {
20        "message": "City must be in valid format!",
21        "type": "user defined",
22        "path": "address.city",
23        "value": "Gwenboroughi"
24      },
25      "kind": "user defined",
26      "path": "address.city",
27      "value": "Gwenboroughi"
28    },
29    "username": {
30      "name": "ValidatorError",
31      "message": "Username field must have length >=4",
32      "properties": {
33        "message": "Username field must have length >=4",
34        "type": "minlength",
35        "minlength": 4
36      }
37    }
38  }
39
```

```

40
41  "email": {
42    "name": "ValidatorError",
43    "message": "Email must be in valid format!",
44    "properties": {
45      "message": "Email must be in valid format!",
46      "type": "user defined",
47      "path": "email",
48      "value": "Sincereapril.biz"
49    },
50    "kind": "user defined",
51    "path": "email",
52    "value": "Sincereapril.biz"
53  },
54  "phone": {
55    "name": "ValidatorError",
56    "message": "Phone Number must be in valid format",
57    "properties": {
58      "message": "Phone Number must be in valid format",
59      "type": "user defined",
60      "path": "phone",
61      "value": "1-712312370-736-8831"
62    },
63    "kind": "user defined",
64    "path": "phone",
65    "value": "1-712312370-736-8831"
66  },
67  "website": {
68    "name": "ValidatorError",
69    "message": "Website must be of valid format",
70    "properties": {
71      "message": "Website must be of valid format",
72      "type": "user defined",
73      "path": "website",
74      "value": "test"
75    }
76  }
77
```

User Schema:

```
const mongoose = require('mongoose');

const UserSchema = new mongoose.Schema({
  name: {
    type: String,
    required: true
  },
  username: {
    required: true,
    type: String,
    minlength: [ 4, 'Username field must have length >=4' ]
  },
  email: {
    required: true,
    type: String,
    validate: {
      validator: function(value) {
        var emailRegex = /^[A-Za-z0-9+_.-]+@[.]+\$/;
        return emailRegex.test(value);
      },
      message: 'Email must be in valid format!'
    }
  },
  address: {
    street: {
      type: String,
      required: true
    },
    suite: {
      type: String,
      required: true
    },
    city: {
      type: String,
      required: true,
      validate: {
        validator: function(value) {
          var alphaSpaceRegex = /^[a-zA-Z ]*$/;
          return alphaSpaceRegex.test(value);
        }
      }
    }
  }
});
```

```
79     required: true,
80     validate: {
81       validator: function(value) {
82         var webRegex = /^((http|https):\/\/)([A-z]+)\.([A-z]{2,})/;
83         return webRegex.test(value);
84       },
85       message: 'Website must be of valid format'
86     }
87   },
88   company: {
89     name: {
90       type: String,
91       required: true
92     },
93     catchPhrase: {
94       type: String,
95       required: true
96     },
97     bs: {
98       type: String,
99       required: true
100    }
101  }
102});
103
104const User = mongoose.model('User', UserSchema);
105module.exports = User;
106
```

```
        },
        zipcode: {
            type: String,
            required: true,
            validate: {
                validator: function(value) {
                    var zipRegex = /^\d{5}-\d{4}$/gm;
                    return zipRegex.test(value);
                },
                message: 'Zipcode must be in valid format'
            }
        },
        geo: {
            lat: {
                type: Number,
                required: true
            },
            lng: {
                type: Number,
                required: true
            }
        }
    },
    phone: {
        type: String,
        required: true,
        validate: {
            validator: function(value) {
                var phoneRegex = /^(\d{1}-\d{3}-\d{3}-\d{4})$/gm;
                return phoneRegex.test(value);
            },
            message: 'Phone Number must be in valid format'
        }
    },
    website: {
        type: String,
        required: true,
        validate: {
            validator: function(value) {
                var webRegex = /^((http|https):\/\/)([A-z]+)\.([A-z]{2,})/;
```

MongoDB collection and documents:

MongoDB Compass - comp3123.okoru.mongodb.net/lab4users_database.users

Connect View Collection Help

Local

15 DBS 27 COLLECTIONS C

HOSTS comp3123-shard-00-00.o... comp3123-shard-00-01.o... comp3123-shard-00-02.o...

CLUSTER Replica Set (atlas-tn0e20-...) 3 Nodes

EDITION MongoDB 5.0.6 Enterprise

Filter your data

lab4users_database

users

local

sample_airbnb

sample_analytics

sample_geospatial

sample_mflix

sample_restaurants

sample_supplies

sample_training

sample_weatherdata

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' } OPTIONS FIND RESET ...

ADD DATA

VIEW

Displaying documents 1 - 10 of 10 < > C REFRESH

Documents

STORAGE SIZE 4.1KB AVG. SIZE 0B TOTAL SIZE 4.1KB AVG. SIZE 4.1KB

DOCUMENTS 0 INDEXES 1

Leanne Graham

Bret

Sincere@april.biz

Ervin Howell

Antonette

Shanna@melissa.tv

Clementine Bauch

Samantha

Nathan@yesenia.net

1-463-123-4447

http://ramiro.info