

Name: Dominic Gopalakrishnan

ID: 101289239

CRN: 49746

Full Stack Development

Lab 4 – Mongoose Validation

Screenshots:

Postman get:

The screenshot shows a Postman interface for a GET request to `http://localhost:3000/users`. The request is successful with a status of 200 OK, a time of 67 ms, and a size of 4.58 KB. The response body is displayed in the 'Pretty' tab, showing a JSON array of user objects. The first object in the array is:

```
{
  "address": {
    "geo": {
      "lat": -43.9569,
      "lng": -34.4618
    },
    "street": "Victor Plains",
    "suite": "Suite 879",
    "city": "Wiskeyburgh",
    "zipcode": "98566-7771"
  },
  "company": {
    "name": "Deckow-Crist",
    "catchPhrase": "Proactive didactic contingency",
    "bs": "synergize scalable supply-chains"
  },
  "_id": "6212dfff9906cf10d01f3568",
  "name": "Ervin Howell",
  "username": "Antonette",
  "email": "Shanna@melissa.tv",
  "phone": "1-991-692-6593",
  "website": "https://anastasia.net",
  "__v": 0
}
```

Error filled postman body:

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": "Gwenborough1",
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  "errors": {
2
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": "Gwenborough1"
24         },
25         "kind": "user defined",
26         "path": "address.city",
27         "value": "Gwenborough1"
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     "email": {
38         "name": "ValidatorError",
39         "message": "Email must be in valid format!",
40         "properties": {
41             "message": "Email must be in valid format!",
42             "type": "user defined",
43             "path": "email",
44             "value": "Sincereapril.biz"
45         },
46         "kind": "user defined",
47         "path": "email",
48         "value": "Sincereapril.biz"
49     },
50     "phone": {
51         "name": "ValidatorError",
52         "message": "Phone Number must be in valid format",
53         "properties": {
54             "message": "Phone Number must be in valid format",
55             "type": "user defined",
56             "path": "phone",
57             "value": "1-712312370-736-8831"
58         },
59         "kind": "user defined",
60         "path": "phone",
61         "value": "1-712312370-736-8831"
62     },
63     "website": {
64         "name": "ValidatorError",
65         "message": "Website must be of valid format",
66         "properties": {
67             "message": "Website must be of valid format",
68             "type": "user defined",
69             "path": "website",
70             "value": "test"

```

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
104 const User = mongoose.model('User', UserSchema);
105 module.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

☆ FAVORITE

HOSTS

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

CLUSTER

Replica Set (atlas-ln0e20-...
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

lab4users_database.users

Documents

DOCUMENTS 0 STORAGE SIZE 4.1KB AVG. SIZE 0B INDEXES 1 TOTAL SIZE 4.1KB AVG. SIZE 4.1KB

Documents Aggregations Schema Explain Plan Indexes Validation

FILTER { field: 'value' }

ADD DATA VIEW

Displaying documents 1 - 10 of 10

REFRESH

```
{ "_id": ObjectId("6212dedd9bee4ece390c2992"),  
  "name": "Leanne Graham",  
  "username": "Bret",  
  "email": "Sincere@april.biz",  
  "address": Object,  
  "phone": "1-770-736-8031",  
  "website": "http://hildegard.org",  
  "company": Object,  
  "__v": 0  
}
```

```
{ "_id": ObjectId("6212dfff0906cf10d81f3558"),  
  "name": "Ervin Howell",  
  "username": "Antonette",  
  "email": "Shanna@melissa.tv",  
  "address": Object,  
  "phone": "1-901-692-6593",  
  "website": "https://anastasia.net",  
  "company": Object,  
  "__v": 0  
}
```

```
{ "_id": ObjectId("6212e0190906cf10d81f355a"),  
  "name": "Clementine Bauch",  
  "username": "Samantha",  
  "email": "Nathan@yesenia.net",  
  "address": Object,  
  "phone": "1-463-123-4447",  
  "website": "http://ramiro.info",  
  "company": Object  
}
```