



# The Internet Folks

## Backend Developer Hiring

### Node Assignment

#### Problem Statement

You run an EdTech-Management SaaS company. On your platform, Schools are registered, along with their Students.

To restrict access, you use Scope-based system, where any Role (ex: Admin, Student, Principal) must have a scope (ex: school-create, user-get) with them to perform any actions on the resources (ex: User, School)

Your task is to create a basic Authentication and CRUD API. The endpoints that you need to create will be given below. The libraries recommended for use are:

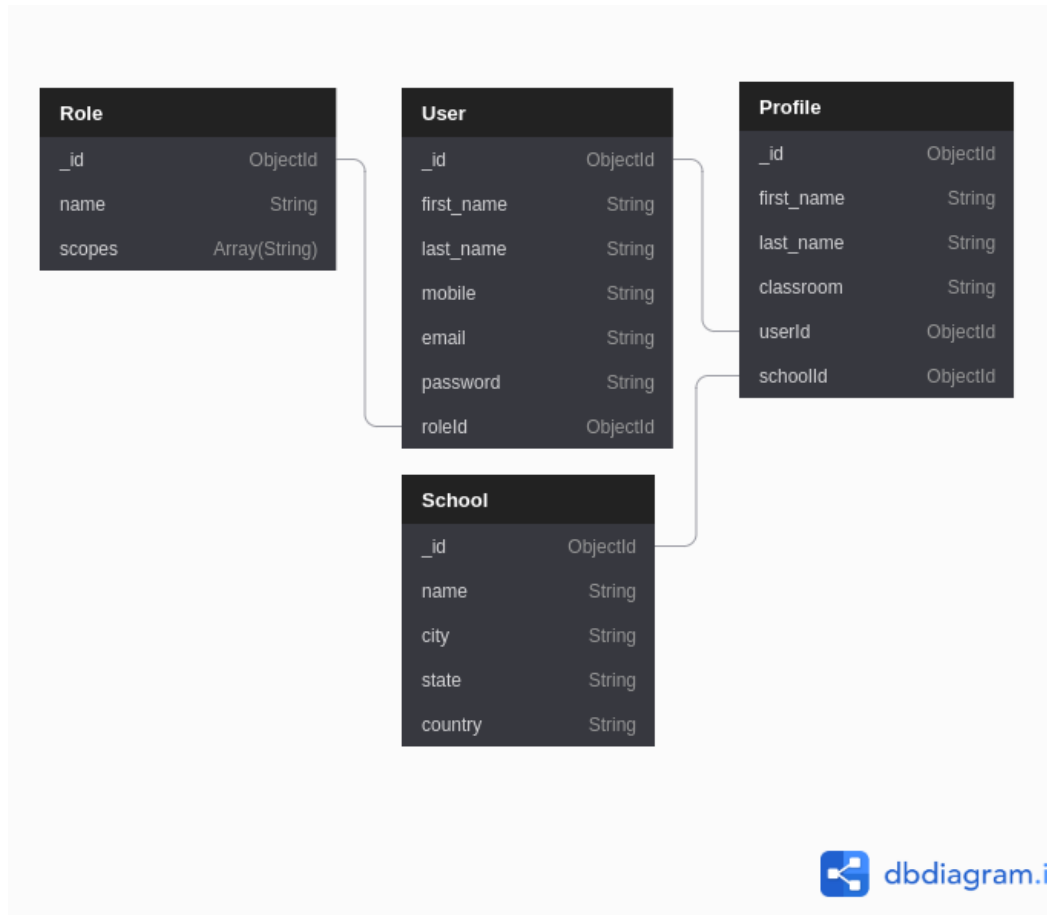
- MongoDB's Native Driver
- Express Validator (<https://www.npmjs.com/package/express-validator>)
- jsonwebtoken (<https://www.npmjs.com/package/jsonwebtoken>)
- Phone (for phone number validation, <https://www.npmjs.com/package/phone>)

**Your task is to create endpoints for:**

- User:
  - Signup
  - Sign in
  - Get Single -> user-get
  - Get List -> user-get
  - Update Single -> user-edit
  - Remove Single -> user-remove
- Profiles:
  - Create -> profile-create
  - Get List -> profile-get
- School
  - Create -> school-create
  - Get All -> school-get
  - Get All Students -> school-get
- Role
  - Create
  - Get List -> role-get
  - Update Single -> role-edit
  - Remove Single -> role-remove

# Database Schema

The basic DB Schema has 2 collections:



DB Diagram: <https://dbdiagram.io/d/6030016dfcdcb6230b20a426>

## User

This is the model for any user.

```
Table User as U {  
  _id ObjectId  
  first_name String  
  last_name String  
  email String [unique]  
  mobile String [unique]  
  password String  
  created DateTime  
  roleId ObjectId [ref: > Role._id, default: null]  
  updated DateTime  
}
```

## Profile

This is the model for any profile.

```
Table Profile {  
  _id ObjectId  
  first_name String  
  last_name String  
  userId ObjectId  
  schoolId ObjectId  
  created DateTime  
  updated DateTime  
}
```

## Role

This is the model for any roles.

```
Table Role {  
  _id ObjectId  
  name String  
  scopes Array(String)  
  created DateTime  
  updated DateTime  
}
```

## School

This is the model for any school.

```
Table School as S {  
  _id ObjectId  
  public_id ObjectId [unique]  
  name String  
  city String  
  state String  
  country String  
  created DateTime  
  updated DateTime  
}
```

# Authentication

When Signing Up, you should store the password after hashing using BCrypt. When Signing in, you should compare the password, and only if it is true, produce a JWT Token.

The generated JWT might look like this:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gR91IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```

The payload of the token will look like this:

```
{
  "user": {
    "_id": "60265e46d560814bb5a8e8bc",
    "first_name": "Ranchoddas",
    "last_name": "Chanchad",
    "email": "rancho@mailinator.com",
    "mobile": "+911234567890",
    "created": "2021-02-12T10:55:51.306Z",
    "updated": null,
    "role": {
      "_id": "60265e46d560814bb5a8e8bc",
      "name": "Manager",
      "scopes": ["user-get", "user-edit", "school-get",
        "school-create", "school-edit"]
    }
  }
}
```



# Endpoints

## User

**POST** /user/signup

Request: (sample)

```
{
  "first_name": "Virus",
  "last_name": "Sahastrabudhhe",
  "email": "virus@mailinator.com",
  "mobile": "+919876543210",
  "password": "iamthebest",
  "roleId": "603927e3fb664daab78eaf31"
}
```

Response: (sample, **success**)

```
{
  "status": true,
}
```

Response: (sample, **error**)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

POST /user/signin

Request: (sample)

```
{
  "email": "virus@mailinator.com",
  "password": "iamthebest"
}
```

Response: (sample, success)

```
{
  "status": false,
  "content": {
    "data": {
      "_id": "60265f8bd2edc95860f1dc3e",
      "first_name": "Viru",
      "last_name": "Sahastrabudhhe",
      "email": "virus@mailinator.com",
      "mobile": "+919876543210",
      "roleId": "603927e3fb664daab78eaf31",
      "created": "2021-02-12T10:55:51.306Z",
      "updated": null
    },
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
  }
}
```

Response: (sample, error)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

GET /user/

Scopes needed: user-get

Response: (sample, success)

```
{
  "status": true,
  "content": {
    "data": [
      {
        "_id": "60265f8bd2edc95860f1dc3e",
        "first_name": "Virus",
        "last_name": "Sahastrabudhhe",
        "email": "virus@mailinator.com",
        "mobile": "+919876543210",
        "roleId": "603927e3fb664daab78eaf31",
        "created": "2021-02-12T10:55:51.306Z",
        "updated": null
      },
      {
        "_id": "60265e46d560814bb5a8e8bc",
        "first_name": "Ranchoddas",
        "last_name": "Chanchad",
        "email": "rancho@mailinator.com",
        "mobile": "+911234567890",
        "roleId": "603927e3fb664daab78eaf31",
        "created": "2021-02-12T10:55:51.306Z",
        "updated": null
      }
    ]
  }
}
```

**PATCH** /user/:\_id

For ex: /user/60265f8bd2edc95860f1dc3e

Scopes needed: user-edit

**Request: (sample)**

```
{
  "first_name": "Virus"
}
```

This will replace the current `first_name` stored in the user's document.

**Response: (sample, success)**

```
{
  "status": true
}
```

**Response: (sample, error)**

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

# School

GET /school/

Scopes needed: school-get

Response: (sample, **success**)

```
{
  "status": true,
  "content": {
    "data": [
      {
        "_id": "60265f8bd2edc95860f1dc3e",
        "public_id": "SCH-U2CK",
        "name": "Imperial College of Engineering",
        "city": "Delhi",
        "state": "Delhi",
        "country": "India",
        "created": "2021-02-12T10:55:51.306Z",
        "updated": null
      }
    ]
  }
}
```

Response: (sample, **error**)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

**GET** /school/:\_id/students

For ex: /school/60265f8bd2edc95860f1dc3e/students

Scopes needed: school-get

Response: (sample, **success**)

```
{
  "status": true,
  "content": {
    "data": [
      {
        "_id": "60265f8bd2edc95860f1dc3e",
        "public_id": "SCH-U2CK",
        "name": "Imperial College of Engineering",
        "city": "Delhi",
        "state": "Delhi",
        "country": "India",
        "students": [
          {
            "_id": "60265f8bd2edc95860f1dc3e",
            "first_name": "Viru",
            "last_name": "Sahastrabudhhe",
            "classroom": "12th",
            "userId": "603e4fe93acac70e1f13cafe",
            "schoolId": "60265f8bd2edc95860f1dc3e"
          },
          {
            "_id": "60265e46d560814bb5a8e8bc",
            "first_name": "Ranchoddas",
            "last_name": "Chanchad",
            "classroom": "11th",
            "userId": "603e4fe00e8cf47311be4442",
            "schoolId": "60265f8bd2edc95860f1dc3e"
          }
        ]
      }
    ]
  }
}
```

Response: (sample, **error**)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

**POST** /school/

Scopes needed: school-create

Request: (sample)

```
{
  "name": "Imperial College of Engineering",
  "city": "New Delhi",
  "state": "Delhi",
  "country": "India"
}
```

Response: (sample, **success**)

```
{
  "status": true
}
```

Response: (sample, **error**)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```

**PATCH** /school/:\_id

For ex: /school/60265f8bd2edc95860f1dc3e

Scopes needed: school-edit

Request: (sample)

```
{
  "city": "Delhi"
}
```

Response: (sample, success)

```
{
  "status": true
}
```

Response: (sample, error)

```
{
  "status": false,
  "errors": [
    {
      "message": "Something went wrong."
    }
  ]
}
```



# Submissions

You can make your submission here: <http://bit.ly/tif-hiring-node-submission>