

Build REST API using NodeJS customer data and store it in an SQL database

This project involves creating a POST API that allows customers to register their personal information, including name, email, mobile number, address, and pin code. The API will validate the input data and securely store the customer details in a SQL database.

Posts		
GET	/posts/	Get Posts
POST	/posts/	Create Posts
GET	/posts/{id}	Get Post
PUT	/posts/{id}	Update Post
DELETE	/posts/{id}	Delete Post

API Endpoint and HTTP Method

The API endpoint will be /customer/register, accepting POST requests.

The appropriate HTTP method for creating a new customer record is POST.

The API should implement authentication and authorization mechanisms to ensure only authorized clients can access and modify customer data.



JSON Input Fields

Required Fields

- First Name
- Last Name
- Email
- Mobile Number (with country code)

Optional Fields

- Address
- Pin Code

Data Types

All fields should be in string format, except for the mobile number which can be a combination of numbers and the "+" symbol.

Validation Requirements

Email Validation

The email address should be validated to ensure it is in a valid format (e.g., john@example.com).

Mobile Number Validation

The mobile number should be validated to ensure it contains a country code and only digits.

Pin Code Validation

The pin code should be validated to ensure it is a 6-digit numeric value.

Other Validations

First name, last name, and address fields should be validated for length and character restrictions.

Database Table Structure

Customer ID	Auto-generated unique identifier
First Name	Text field
Last Name	Text field
Email	Text field
Mobile Number	Text field
Address	Text field
Pin Code	Text field

Stored Procedure Implementation

Ensure all required fields are present and meet the validation criteria.

Use an auto-increment or GUID-based mechanism to generate a unique identifier for the new customer record.

Insert the validated customer data into the "Customer Details" table using the generated Customer ID.



API Response Handling



Success

If the customer registration is successful, return a 201 Created status code and the newly generated Customer ID.



Error

If the input data is invalid or there is an issue with the database operation, return a 400 Bad Request status code and a detailed error message.



Logging

Implement robust logging mechanisms to track API usage, errors, and debugging information.

Screenshots

```
index.js
1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const mysql = require('mysql2');
4  const { check, validationResult } = require('express-validator');
5
6  const app = express();
7  const port = 3000;
8
9  app.use(bodyParser.json());
10
11
12  const db = mysql.createConnection({
13    host: 'localhost',
14    user: 'root',
15    password: 'mysql',
16    database: 'dush_cust_db'
17  });
```

Code

Index.js connection

```
localhost:3000/api/customers
Pretty-print [x]
[
  {
    "CustomerID": 1,
    "FirstName": "K. Dushyant",
    "LastName": "Reddy",
    "Email": "k.dushyantreddy@gmail.com",
    "MobileNumber": "+919150866039",
    "Address": "Hyderabad",
    "Pincode": "500085"
  },
  {
    "CustomerID": 2,
    "FirstName": "Raju",
    "LastName": "R",
    "Email": "raju@gmail.com",
    "MobileNumber": "+911234567890",
    "Address": "Thane",
    "Pincode": "400080"
  }
],
```

Frontend

API JSON details

```
mysql> show databases;
+-----+
Database
+-----+
dush_cust_db
information_schema
mysql
performance_schema
sys
5 rows in set (0.11 sec)

mysql> use dush_cust_db;
Database changed
mysql> show tables;
+-----+
Tables_in_dush_cust_db
+-----+
customerdetails
1 row in set (0.03 sec)

mysql> select * from customerdetails
+-----+
CustomerID | FirstName | LastName | Email | MobileNumber | Address | Pincode
+-----+
1 | K. Dushyant | Reddy | k.dushyantreddy@gmail.com | +919150866039 | Hyderabad | 500085
2 | Raju | R | raju@gmail.com | +911234567890 | Thane | 400080
3 | Raju | K | k.raju@gmail.com | +911234567890 | Mumbai | 400001
4 | Ajay | A | ajay@gmail.com | +911234567890 | Mumbai | 400001
4 rows in set (0.02 sec)
```

Backend

MySQL database

Conclusion

To create a POST API that accepts JSON input for first name, last name, email, mobile number (with country code), address, and pin code. Implement validation for email, mobile number, and pin code. Upon receiving a valid request, store the data in a SQL database using a stored procedure, with a table named "Customer Details". The table will include fields for a unique Customer ID (auto-generated), first name, last name, email, mobile number, address, and pin code. The API will return a success response on successful insertion or a failure response if validation fails or the insertion is unsuccessful.

Done by: **K. Dushyant Reddy**

+91 9150866039

 Goo... 

 GitHub 

 www.l... 

