

CapitalLookupAPI

Front end provides an intuitive interface for users to find the capital of a country. It features responsive design, input validation, and error handling, making it user-friendly and functional. This setup seamlessly connects to your Node.js API, fetching data dynamically based on user input.

Structure:

(Backend / api)

MONGODB + (NODE Js and Express Js) connection

```
let mongoose = require('mongoose');
mongoose.connect("mongodb://127.0.0.1:27017/dbapi");
let apischema = mongoose.Schema({
  country: String,
  capital: String
});
module.exports = mongoose.model('api',apischema);

//api build back end code

let express = require('express');
let app = express();
let api = require('./dbapi.js');
let cors = require('cors');

app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

app.get('/:country', async (req,res)=>{
  const country = req.params.country;
  let apidata = await api.findOne({"country": country.toUpperCase()},{_id:
false});
  if(apidata){
    console.log(apidata);
    res.json(apidata);
  }
  else{
    res.status(404).json({msg:"mongodb data fetch error"});
  }
});
```

```

    }
    //res.send(apidata);
  });

  app.listen(3003,(err)=>{
    console.log("Running...");
  });

```

Working:

This API is a simple Node.js application that connects to a MongoDB database to fetch information about countries and their capitals. It uses Express.js to handle HTTP requests and responses.

Breakdown of the Code

1. MongoDB Connection and Schema Definition

```

let mongoose = require('mongoose');

mongoose.connect("mongodb://127.0.0.1:27017/dbapi", { useNewUrlParser:
true, useUnifiedTopology: true })
  .then(() => console.log("MongoDB connected..."))
  .catch(err => console.error("MongoDB connection error:", err));

let apischema = new mongoose.Schema({
  country: { type: String, required: true },
  capital: { type: String, required: true }
});

module.exports = mongoose.model('api', apischema);

```

- **Connection**: You establish a connection to a MongoDB instance running on `localhost` at port `27017`. You're connecting to a database named `dbapi`.
- **Schema Definition**: You define a schema called `apischema` that describes the structure of the documents in your MongoDB collection:
 - `country`: A string representing the name of the country (required).
 - `capital`: A string representing the capital city of the country (required).
- **Model Export**: You export the model created with Mongoose, allowing you to interact with the corresponding MongoDB collection.

2. Express.js API Setup

```

let express = require('express');
let app = express();
let api = require('./dbapi.js');
let cors = require('cors');

app.use(cors());
app.use(express.json());
app.use(express.urlencoded({ extended: true }));

```

- **Express Setup**: You create an Express application instance (`app`) and import your MongoDB model (`api`).

- **Middleware**:

- `cors()`: Enables Cross-Origin Resource Sharing, allowing your API to be accessed from different domains.

- `express.json()`: Parses incoming requests with JSON payloads.

- `express.urlencoded()`: Parses incoming requests with URL-encoded payloads.

3. GET Endpoint to Fetch Country Data

```

app.get('/:country', async (req, res) => {
  const country = req.params.country.toUpperCase();

  try {
    let apidata = await api.findOne({ "country": country }, { _id: false });

    if (apidata) {
      console.log(apidata);
      res.json(apidata);
    } else {
      res.status(404).json({ msg: "Country not found in the database." });
    }
  } catch (error) {
    console.error("Database fetch error:", error);
    res.status(500).json({ msg: "Internal server error" });
  }
});

```

- **Route Definition**: You define a GET route that responds to requests at the path `/:country`, where `/:country` is a URL parameter.

- **Parameter Handling**: The parameter is converted to uppercase to ensure case-insensitive matching with the database.
- **Database Query**: Using `api.findOne()`, you query the database for a document that matches the provided country. You exclude the `_id` field from the response.
- **Response Handling**:
 - If a document is found, it logs the data and returns it as a JSON response.
 - If no document is found, it sends a 404 response with an appropriate message.
 - If there's an error while querying the database, it logs the error and returns a 500 status code indicating an internal server error.

4. Starting the Server

```
app.listen(3003, (err) => {
  if (err) {
    console.error("Error starting the server:", err);
  } else {
    console.log("Server is running on port 3003...");
  }
});
```

- start the Express server on port `3003`. If the server starts successfully, it logs a message to the console.

Summary

This API provides a way to fetch country data from a MongoDB database based on the country name provided in the request. It handles requests, queries the database, and returns the results or appropriate error messages. This structure is quite efficient for a simple API and can be expanded further if needed!

Strengths:

Basic Structure: The overall structure is clear and well-organized. You've separated the database schema and the API logic, which is good for maintainability.

CORS Support: Including CORS is great for allowing cross-origin requests, which is important for modern web applications.

Use of Async/Await: Utilizing async/await for handling asynchronous database operations improves readability and error handling.

JSON Support: By using `express.json()` and `express.urlencoded()`, you're correctly set up to handle JSON and form data.

MONGO DB data set

//Example only (All of the countries in the world are available)

```
[{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b8e"
  },
  "country": "AFGHANISTAN",
  "capital": "KABUL"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b8f"
  },
  "country": "ALBANIA",
  "capital": "TIRANA"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b90"
  },
  "country": "ALGERIA",
  "capital": "ALGIERS"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b91"
  },
  "country": "ANDORRA",
  "capital": "ANDORRA LA VELLA"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b92"
  },
  "country": "ANGOLA",
  "capital": "LUANDA"
}
```

```
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b93"
  },
  "country": "ANTIGUA AND BARBUDA",
  "capital": "ST. JOHN'S"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b94"
  },
  "country": "ARGENTINA",
  "capital": "BUENOS AIRES"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b95"
  },
  "country": "ARMENIA",
  "capital": "YEREVAN"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b96"
  },
  "country": "AUSTRALIA",
  "capital": "CANBERRA"
},
{
  "_id": {
    "$oid": "66ffc8ef6cfb6afeb1337b97"
  },
  "country": "AUSTRIA",
  "capital": "VIENNA"
},
{
  "_id": { "$oid": "66ffc8ef6cfb6afeb1337b98"
  },
  "country": "AZERBAIJAN",
  "capital": "BAKU"
}
]
```