NEWS

API

Student
Mohammed Riyaan

Index

Page No	Title
3	About News API
5	Tech Stack
7	Proposed Solution
10	Demo and Installation of News API

About Project News API

Application Programming Interface (API)

API stands for Application Programming Interface. In simple words, an API is like a messenger that allows different software applications to communicate and interact with each other. It defines a set of rules and protocols that determine how one software application can access and use the functionalities or data of another application.

An API acts as an intermediary that enables different software applications to exchange information, request services, or perform actions. It provides a simplified and standardized way for developers to access the functionalities or data of another application without needing to understand the internal workings of that application.

NewsAPI

A news API, in simple words, is a service that provides developers with access to a collection of news articles, headlines, and related information from various sources such as newspapers, blogs, or news websites. It allows developers to retrieve and integrate news content into their own applications, websites, or services.

Using a news API, developers can make requests for specific news articles or retrieve a list of articles based on certain criteria, such as keywords, categories, or dates. The API responds with the requested news data, usually in a structured format like JSON (JavaScript Object Notation), which developers can then process and display as desired.

News APIs are beneficial because they save developers from the task of manually gathering news information from multiple sources. Instead, they can rely on the API to provide them with a stream of up-to-date news content that they can utilize in their applications. This enables developers to create news aggregators, personalized news apps, or any other application that requires access to the latest news information, without having to directly manage and maintain a comprehensive news database themselves.

Tech stack used in News API

Backend - Server

1. Programming Languages - JavaScript
2. Runtime Environment - Node JS
3. BackEnd Framework – ExpressJS

Client Tools

PostMan - Postman is a popular client tool used for testing and interacting with web APIs (Application Programming Interfaces). It provides a user-friendly interface that allows developers to make requests to APIs, inspect responses, and automate API testing and documentation.

NPM Packages

1. express:- The Express npm package is a fast, unopinionated, and minimalist web application framework for Node.js. It provides a robust set of features for building web applications and APIs. Express is one of the most popular and widely used frameworks in the Node.js ecosystem due to its simplicity and flexibility.

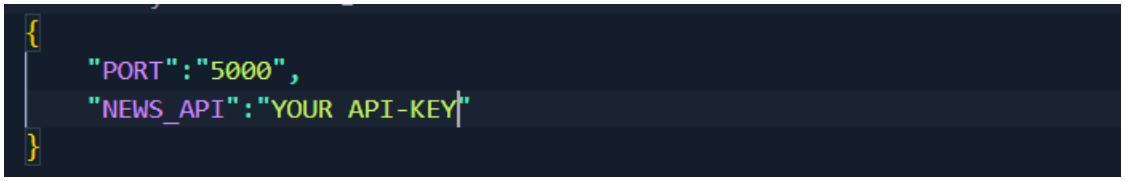
2. config:- The "config" npm package is a popular configuration management library for Node.js applications. It provides a convenient way to define and access configuration values for different environments (e.g., development, production, staging)

3. axios:- The "axios" npm package is a popular JavaScript library used for making HTTP requests from both browsers and Node.js applications. It provides a simple and intuitive API for handling asynchronous operations and interacting with APIs.

4. Proposed Solution

Folder Structure

1) config :- We are creating a config directory in your project's root directory. This is where you'll store your configuration files . Inside the config directory, create a configuration file for each environment or configuration profile you want to define. For example, you can create default.json for common settings. Here's an ex of default.json



2) controllers :- The controllers folder contains the business logic or handlers for each route. It separates the route handling from the application's core logic.

3) node_modules :- This folder contains the installed npm packages and their dependencies.

4) .gitignore :- The purpose of gitignore files is to ensure that certain files not tracked by Git remain untracked.

5) app.js :- This file serves as the entry point of the application where you set up the Express server, configure middleware, and define routes.

6) package-lock.json :- package-lock. json is a file that is automatically generated by npm when a package is installed. It records the exact version of every installed dependency, including its sub-dependencies and their versions.

7) package.json :- The package.json file contains information about the project, including its dependencies, scripts, and metadata.

8) README.md :- This file typically includes documentation about the project, providing information on how to set it up, run it, and any other relevant details.

Procedure

Installing express , config and axios packages

Importing packages

```
import express from "express"
import config from "config"
import axios from "axios"
```

We are declaring PORT and newsAPI in which the values used in config file are stored

When you execute const app = express(), you are creating a new Express application by invoking the express() function. By assigning the returned application object to the variable app (using the const keyword), you can use the app variable to configure and define routes, middleware, and other aspects of your Express application.

```
const PORT = config.get("PORT")
const newsAPI = config.get("NEWS_API")
const app = express()
```

We are defining the first route for the root URL (/) using the app.get() method. The callback function (req, res) => {...} is executed when an HTTP GET request is made to the root URL. In this case, it sends the response in json format {message:"Home Page"} back to the client with a status code (200).

```
app.get("/",(req,res)=>{
    res.status(200).json({message:"Home Page"})
})
```

The second route sets up a route with the path /:country, where :country is a parameter representing the desired country news (e.g., united states[us], india[in], etc.). Inside the route handler, an asynchronous function is defined using async and await. It uses Axios to make an HTTP GET request to the News API, passing the desired country code [in, us]

and the API key in the URL. The response is then extracted and the articles are stored in data variable . Then to reperesent the top 5 articles we are using slice and map method and then stored in articles variable.

Finally , the response is sent to the client in a json format using the value of articles with status code (200).

If there is an error during the API request, an error message (“Internal Server Error”) along with 500 HTTP status code is sent as the response in json format.

```
app.get("/:country",async(req,res)=>{
    try {
        const country = req.params.country
        const newsData = await axios.get(`https://newsapi.org/v2/top-headlines?country=${country}&apiKey=${newsAPI}`)
        let data = newsData.data.articles
        let articles = data.slice(0,5).map((x)=>(x.title))
        res.status(200).json({"The Top 5 Articles are" : articles})
    } catch (error) {
        res.status(500).json({message:"Internal Server Error"})
    }
})
```

If we enter the invalid route then the response should be “Invalid Route” in json format along with status code (404).

```
app.use((req,res)=>{
    res.status(404).json({message:"Invalid Route"})
})
```

The below code starts the server and listens on port . Once the server is up and running, the callback function (req, res) => {...} is executed, which logs the message `Server Listening At Port ${PORT}` to the console.

```
app.listen(PORT,()=>{
    console.log(`Server Listening At Port ${PORT}`)
})
```

Installation of News API

GitHub Repository: https://github.com/mohdriyaan/News-API

To clone and Install packages:

```
riyaan@riyaan:~$ git clone git@github.com:mohdriyaan/News-API.git
Cloning into 'News-API'...
remote: Enumerating objects: 17, done.
remote: Counting objects: 100% (17/17), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 17 (delta 1), reused 8 (delta 0), pack-reused 0
Receiving objects: 100% (17/17), 12.52 KiB | 4.17 MiB/s, done.
Resolving deltas: 100% (1/1), done.
```

To Run Project

cd News-API

npm i

npm start

Output :- After Executing The Server Starts Running At the Designated Port.

```
riyaan@riyaan:~/News-API$ npm start

> news-api@1.0.0 start
> node app.js

Server Listening At Port 5000
```

After starting the server , you can access your application in a web browser or client tool such as PostMan by navigating to https://YOUR-WIFI-IP:5000. In this case we are using PostMan.

The Output Of First Route http://192.168.1.13:5000 Will Be :-

The Output Of Second Route http://192.168.1.13:5000/in Will Be :-